Divide and Conquer: Recursive Likelihood Function Integration for Hidden Markov Models with Continuous Latent Variables

Reich, Gregor

Abstract: This paper develops a method to efficiently estimate hidden Markov models with continuous latent variables using maximum likelihood estimation. To evaluate the (marginal) likelihood function, I decompose the integral over the unobserved state variables into a series of lower dimensional integrals, and recursively approximate them using numerical quadrature and interpolation. I show that this procedure has very favorable numerical properties: First, the computational complexity grows linearly in the number of periods, making the integration over hundreds and thousands of periods feasible. Second, I prove that the numerical error accumulates sublinearly in the number of time periods integrated, so the total error can be well controlled for a very large number of periods using, for example, Gaussian quadrature and Chebyshev polynomials. I apply this method to the bus engine replacement model of Rust [Econometrica 55(5): 999–1033] to verify the accuracy and speed of the procedure in both actual and simulated data sets.

DOI: https://doi.org/10.1287/opre.2018.1750

Divide and Conquer: Recursive Likelihood Function Integration for Hidden Markov Models with Continuous Latent Variables

Gregor Reich*

Dept. of Business Administration
University of Zurich
gregor.reich@uzh.ch

November 7, 2017

Abstract

This paper develops a method to efficiently estimate hidden Markov models with continuous latent variables using maximum likelihood estimation. To evaluate the (marginal) likelihood function, I decompose the integral over the unobserved state variables into a series of lower dimensional integrals, and recursively approximate them using numerical quadrature and interpolation. I show that this procedure has very favorable numerical properties: First, the computational complexity grows linearly in time, which makes the integration over hundreds and thousands of periods well feasible. Second, I prove that the numerical error is accumulated sub-linearly over time; consequently, using highly efficient and fast converging numerical quadrature and interpolation methods for low and medium dimensions, such as Gaussian quadrature and Chebyshev polynomials, the numerical error can be well controlled even for very large numbers of periods. Lastly, I show that the numerical convergence rates of the quadrature and interpolation methods are preserved up to a factor of at least 0.5 under appropriate assumptions. I apply this method to the bus engine replacement model of Rust [*Econometrica*, 55 (5): 999–1033, (1987)]: first, I estimate the model using the original dataset; second, I verify the algorithm's ability to recover the parameters in an extensive Monte Carlo study with simulated datasets.

Subject classifications: Economics: Econometrics; Statistics: Estimation; Dynamic Programming: Applications.
Area of review: Computational Economics.

*I am heavily indebted to my advisor Karl Schmedders as well as to Ken Judd, John Rust, and Che-Lin Su for their support and guidance in this project. I would also like to thank Greg Crawford, Philipp Eisenhauer, Katharina Erhardt, Dennis Kristensen, János Mayer, Bertel Schjerning, Ole Wilms, and seminar audiences at the University of Chicago, Hoover Institution Standford University, University of Zurich, and the 68th European Meeting of the Econometric Society for helpful comments and suggestions. Finally, I thank Dave Brooks for editorial comments on the manuscript. Earlier versions of this paper circulated as "Divide and Conquer: A New Approach to Dynamic Discrete Choice with Serial Correlation" (2013) and "Divide and Conquer: Recursive Likelihood Function Integration for Dynamic Discrete Choice Models with Serially Correlated Unobserved State Variables" (2016).

1 Introduction

This paper develops a method to efficiently estimate hidden Markov models with continuous latent variables using maximum likelihood estimation. To evaluate the (marginal) likelihood function, I decompose the integral over the unobserved state variables into a series of lower dimensional integrals, and recursively approximate them using numerical quadrature and interpolation. I show that this procedure has very favorable numerical properties: First, the computational complexity grows linearly in time, which makes the integration over hundreds and thousands of periods well feasible. Second, I prove that the numerical error is accumulated sub-linearly over time; consequently, using highly efficient and fast converging numerical quadrature and interpolation methods for low and medium dimensions, such as Gaussian quadrature and Chebyshev polynomials, the numerical error can be well controlled even for very large numbers of periods. Lastly, I show that the numerical convergence rates of the quadrature and interpolation methods are preserved up to a factor of at least 0.5 under appropriate assumptions. I apply this method to the bus engine replacement model of Rust [*Econometrica*, 55 (5): 999–1033, (1987)]: first, I estimate the model using the original dataset; second, I verify the algorithm's ability to recover the parameters in an extensive Monte Carlo study with simulated datasets.

An important application of hidden Markov models within economics are the dynamic discrete choice models (DDCMs). While plenty of other uses exist—inside and outside of economics (see, for example the classic textbook of Elliott et al., 2008)—this paper's focus is on DDC modeling of economic decision making, which has become a very popular tool in the last three decades: First, many (individual) economic decisions we actually can observe are in fact discrete in nature, for example the choice of a brand or medical treatment. Second, the underlying utility maximization problem of the agents is often dynamic in nature: decisions made today not only influence today's payoffs, rather they also influence future decisions and payoffs. By capturing these key facts, DDCMs have a wide range of uses; for the pioneering papers see, for example, Miller (1984); Pakes (1986); Rust (1987); Wolpin (1984). For recent surveys see Aguirregabiria and Mira (2010); Keane et al. (2011).

The majority of contributions to the literature on the estimation of DDCMs make strong distributional assumptions about the errors and other unobserved state variables. Probably the most prominent example is extreme value type I $EV1$ iid distributed errors; obviously implied by the $EV1$ iid assumption, but usually stated explicitly by a conditional independence assumption (CI), the errors are assumed to be serially uncorrelated. However, there exists a wide consensus that these assumptions are not made based on the existence of much empirical evidence, but rather for numerical tractability: serial independence—alongside with other distributional assumptions—induce closed form solutions to potentially high dimensional integrals that arise in the solution to the dynamic optimization problem and in the choice probabilities in the likelihood function. These closed form solutions go back to the work of McFadden (1974, 1981) and Rust (1987). If, however, no closed form solutions exist, it is common understanding that the likelihood function is hard to compute:

> "the likelihood function for a DDCM can be thought of as an integral over latent variables (the unobserved state variables). If the unobservables are serially correlated,

computing this integral is very hard." (Norets, 2009)

This conclusion follows from the fact that the integral over serially correlated errors really has dimensionality proportional to the time horizon of the data, which itself can be arbitrarily large.

While relaxing the $EV1$ error assumption has attracted some attention—for example, Larsen et al. (2012) test the statistical significance of allowing for more general distributions in the Rust (1987) model—several papers have developed integrated methods to estimate models without the CI assumption, thus allowing for a general notion of serially correlated unobserved state variables. Among those are the expectation–maximization algorithm based on conditional choice probability estimation of Arcidiacono and Miller (2011), the particle filter method of Blevins (2016), and the Markov chain Monte Carlo approaches of Norets (2009, 2012). Apart from those, several papers use Monte Carlo (MC) integration to directly approach the integration over the unobserved state variables; among them are the simulation and interpolation method of Keane and Wolpin (1994), the Patent model of Pakes (1986) which is considered one of the pioneering DDC models, and the application of Gaussian quadrature and interpolation as discussed in Stinebrickner (2000). The application of MC integration is motivated by the fact that the variance of the MC estimate of the integral does not depend on the dimension of the integral and thus not on the time horizon of the data.

The approach followed in this paper is quite different by identifying and exploiting the structure that is present in the integral over the unobserved state variables in the (marginal) likelihood function: Given the serial dependence of the unobserved state variables is Markov, the time structure allows the high dimensional integral over the time horizon to be decomposed and rewritten as a sequence of low dimensional integrals. Then, I can recursively approximate this sequence to high accuracy, using highly efficient approximation schemes for low dimensional integrals, such as Gaussian quadrature, and interpolate this approximation to iterate over the time dimension.[1] While it is straightforward to see that the computational complexity of computing this integral is linear in the time dimension, one of the main contributions of this paper is the analysis of the numerical properties of this method, which I call "recursive likelihood function integration" (RLI): First, the accumulation of the numerical error from repeatedly approximating integrals and functions by quadrature and interpolation, respectively, is investigated; I find that while error accumulation is present, it grows only at a sub-linear rate (in the worst case), which can easily be compensated for. Second, the convergence rate of the method is derived in terms of the convergence rates of the quadrature and the interpolation methods used; I find that under some general assumptions, the convergence rates of the employed methods are preserved at least up to a factor of at least 0.5. Lastly, I formulate generic assumptions on the continuous state hidden Markov models that make the convergence results of the RLI method applicable.

While the focus of the paper is to approximate the likelihood function of hidden Markov models, solving the DDC model usually also requires substantial numerical work, unless a two-step estimator in the sense of Hotz and Miller (1993) is used (also see Arcidiacono and Ellickson,

[1] Recursive computation of the likelihood function for serially correlated unobserved Markov states is not a new idea in general. However, to the best of my knowledge, its application has been limited to discrete state spaces, and therefore with no need for numerical quadrature or function approximation; see, for example, Cosslett and Lee (1985) for the estimation of models with Markov regime switching. Recently, Connault (2016) applied this idea to compute the likelihood function of dynamic discrete choice models with discrete unobserved state variables, as well as discussing identification in such setups, which he refers to as "Hidden Rust Models".

2011, for a recent study on two-step estimators), or unless the solution of the model is combined with its estimation in a Bayesian framework, as done by Imai et al. (2009); Norets (2009). Several approaches to value function approximation have been proposed (see, for example, Cai and Judd, 2013; Judd, 1998; Rust, 1996), and to stay flexible and generic I use interpolation over an adaptively refined grid, as proposed by Grüne and Semmler (2004); for the computation of the expectation over the value, I use Gaussian quadrature as was first proposed and successfully implemented in the context of DDCMs with serially correlated unobserved state variables by Stinebrickner (2000). Finally, I solve the maximum likelihood problem using a nested fixed point algorithm (NFXP; Rust, 1987), which is interconnected with the grid refinement process of the expected value function approximation.

As an application, I estimate the bus engine replacement model of Rust (1987) with serially correlated errors. One motivation for serial correlation in this model is a test for misspecification from the original paper, which leads to the following conclusion:[2]

> "for groups 1, 2, and 3 and the combined groups 1–4 there is strong evidence that (CI) does not hold. The reason for rejection in the latter cases may be due to the presence of 'fixed-effects' heterogeneity which induces serial correlation in the error terms." (Rust, 1987)

Testing for statistical significance of serially correlated errors I find that in some subsamples of the original dataset I can reject serially uncorrelated errors. Also, the parameter estimates vary substantially; their relative sizes however are rather stable.

The remainder of this paper is organized as follows: Section 2 first presents a motivating example by extending the bus engine replacement model of Rust (1987) to feature serially correlated errors (Section 2.1). Second, its solution and estimation procedure is discussed, and a simple version of the recursive likelihood function integration algorithm is derived (Section 2.2). Section 3 first introduces rigorous definitions and assumptions on the integration and interpolation problems as well as examples of solution methods (Sections 3.1 and 3.2). Second, the recursive likelihood function integration method is analyzed with respect to accumulation of numerical error, convergence speed (Section 3.3), and applicability to general continuous state hidden Markov models (Section 3.4). Section 4 presents the estimation results for the bus engine replacement model with serially correlated errors using RLI. Section 5 concludes and states the agenda for future research.

2 A Motivating Example

This section will introduce the topic of estimating dynamic discrete choice models with serially correlated unobserved state variables by presenting a popular example, extending it to feature

[2] One can also think of serial correlation as a "generic feature" in this context: In optimal stopping problems, such as the bus engine replacement model, the replacement decision is expected to happen rarely. If the explanatory power of the model in terms of observed states is low, the probability of stopping is small for all possible observed states. Thus, the observed decisions are mostly driven by tail events of the unobserved state variables. However, this fact contradicts the assumption that decisions are modeled to be dynamic, because in a model without serial correlation, these events are unforeseeable, single period shocks. With the introduction of serial correlation, these shocks have persistent effects, which can be anticipated by the agent. For example, a jump in maintenance costs still comes as a surprise to the agent, but—once incurred—its effect on future periods can influence decisions to a large extent.

serially correlated errors, and sketch a method how to finally solve and estimate the model. While the proposed recursive likelihood function integration method is motivated and outlined in full detail in this section, its theoretical properties (speed of convergence and error analysis) as well as its scope of applicability are discussed rigorously in Section 3. (Since the main focus of this paper is the computation of the likelihood function, I defer parts of the solution details to the appendix.)

2.1 The Bus Engine Replacement Model

In the bus engine replacement model of Rust (1987), an agent repeatedly makes decisions about the maintenance of a fleet of buses: Each period, he observes the state of each of the buses, including mileage, damage, signs of wear, etc. Based on these observations, he decides whether to do regular maintenance work only, or a general overhaul; the latter is usually referred to as a replacement of the engine. While the engine replacement causes a fixed cost of RC plus some random component, the cost of regular maintenance is a function $c(\cdot)$ that is increasing in the current mileage state, plus some random component.

Formally, the agent faces single period costs (or negative utility) for each individual bus

$$u_\theta(i, x_t) + \varepsilon_t(i)\,, \quad u_\theta(i, x_t) = \begin{cases} -RC & \text{if } i = 1 \\ -c(x_t, \theta_1) & \text{if } i = 0 \end{cases} \tag{1}$$

where i is the decision variable, with $i = 1$ indicating engine replacement, and $i = 0$ regular maintenance; $\varepsilon_t(i)$ is a random utility component that is observed by the agent for all possible choices before making the actual decision; x_t is the mileage of the individual bus at time t, which is reset to 0 after an engine replacement. The replacement cost RC, as well as the cost function parameter θ_1, are both parameters to be estimated. The maintenance cost function is assumed to be of the form $c(x_t, \theta_1) = 0.001\,\theta_1\,x_t$. From the econometrician's point of view, mileage at the time of decision and the decision itself are observable for each bus and each time period. The random utility component however is only observable to the agent, but not to the econometrician; consequently, it is often referred to as the unobserved state variable.

For the agent, the decision problem is how long to run a bus with regular maintenance only, with increasing costs induced by increasing mileage, and when to replace its engine, thus facing the one-time replacement cost, but at the same time reducing the maintenance costs in the future because mileage is reset to 0. Assuming that the agent behaves dynamically optimally, the Bellman equation defines the value per bus as a function of its mileage state and the random utility components

$$V_\theta(x_t, \varepsilon_t) = \max_{i \in \{0,1\}} \{u_\theta(i, x_t) + \varepsilon_t(i) + \beta \mathbb{E}[V_\theta(x_{t+1}, \varepsilon_{t+1}) | i, x_t, \varepsilon_t; \theta]\}. \tag{2}$$

The conditional expected continuation value in (2) is defined by

$$\mathbb{E}[V_\theta(x_{t+1}, \varepsilon_{t+1}) | i, x_t, \varepsilon_t; \theta] = \int V_\theta(x_{t+1}, \varepsilon_{t+1})\, p_{x\varepsilon}(x_{t+1}, \varepsilon_{t+1} | i, x_t, \varepsilon_t; \theta) d(x_{t+1}, \varepsilon_{t+1}) \tag{3}$$

with subscript θ denoting the dependence of the value function on the parameter values RC

and θ_1, and where the integration limits are ignored for better readability. $p_{x\varepsilon}$ is the conditional joint density function of the state variable process.

The original model makes the following conditional independence (CI) assumption regarding the joint distribution of the state variables:

$$p_{x\varepsilon}(x_{t+1}, \varepsilon_{t+1}|i, x_t, \varepsilon_t; \theta) = \tilde{p}_{\varepsilon|x}(\varepsilon_{t+1}|x_{t+1}; \theta) p_x(x_{t+1}|i, x_t; \theta) \tag{4}$$

Assumption (4) ensures that (i) the mileage state transition is—conditional on the decision i—independent of the random utility component, and (ii) that the random utility components are serially uncorrelated. If the CI assumption holds, and if moreover the random utility components $\varepsilon(i)$ are distributed extreme value type I ($EV1$) iid, the integral in (3) has a closed form solution. However, in order to allow for serial correlation in ε, while keeping (i), I assume

$$p_{x\varepsilon}(x_{t+1}, \varepsilon_{t+1}|i, x_t, \varepsilon_t; \theta) = p_{\varepsilon|x}(\varepsilon_{t+1}|\varepsilon_t, x_{t+1}; \theta) p_x(x_{t+1}|i, x_t; \theta) \tag{5}$$

Note that assumption (5) allows the transition process of the mileage state, $p_x(x_{t+1}|i, x_t; \theta)$, to be estimated independently from the other model parameters—as in the original model.[3] I use discretized mileage, and thus the integral over future mileage states in (3) becomes a sum:

$$\mathbb{E}[V_\theta(x_{t+1}, \varepsilon_{t+1})|i, x_t, \varepsilon_t; \theta] = \sum_{x_{t+1}} \int V_\theta(x_{t+1}, \varepsilon_{t+1})\, p_{\varepsilon|x}(d\varepsilon_{t+1}|\varepsilon_t, x_{t+1}; \theta)\, p_x(x_{t+1}|i, x_t; \theta) \tag{6}$$

A choice for serial correlation in the unobserved state variables that is frequently used in the literature is the $AR(1)$ process. More specifically, I define

$$\begin{aligned} \varepsilon_t(0) &= \rho\varepsilon_{t-1}(0) + \tilde{\varepsilon}_t(0), \quad \tilde{\varepsilon}_t(0) \text{ iid} \\ \varepsilon_t(1) &= \qquad\qquad\quad \tilde{\varepsilon}_t(1), \quad \tilde{\varepsilon}_t(1) \text{ iid} \end{aligned} \tag{7}$$

and $q(\cdot)$ as the probability density function of $\tilde{\varepsilon}_t(i)$ with zero mean. Note that ρ is an additional parameter of the estimation; furthermore, I assume that $\varepsilon_0(i)$ is distributed with density $q(\cdot)$. Thus, I only assume the random utility component of regular maintenance to be serially correlated. It is important to note that definition (7) nests the original model for $\rho = 0$, and the density function $q(\cdot)$ being extreme value type 1, $EV1$.[4] Moreover, I consider two variants for each density function: The first variant uses the "standard" form of the distribution—like the standard normal distribution with mean zero and variance one—, whereas the second normalizes the distribution of the innovation $\tilde{\varepsilon}$ such that the resulting $AR(1)$ process has zero mean and constant variance (thus independent of ρ), which is achieved by setting the location and scale parameters accordingly (see Section 4.1 for details).

Given that mileage state x_t and decision i_t are observable for all buses, but random utility components ε_t are not, the aim is to estimate this model's parameter $\theta = \{\theta_1, RC, \rho\}$, given the

[3] Since one can estimate the mileage transition process $p_x(x_{t+1}|i, x_t; \theta)$—referred to as parameter θ_3 in the original model—independently from $\theta = \{\theta_1, RC, \rho\}$, and moreover, since it is exactly the same as in Rust (1987) (because it is not affected by the serial correlation in the unobserved state variables) I ignore this aspect of the bus engine replacement model in the remainder of this paper.

[4] I silently assume that after a replacement, the series of serially correlated unobserved states is reset to its mean, 0. Thus, $\varepsilon(i)$ in the first period after an engine replacement is distributed according to density $q(\cdot)$ again.

data $\{x_t, i_t\}_{t=0}^T$, by maximum likelihood estimation.

2.2 Computation and Estimation

This subsection develops a numerical method to estimate the bus engine replacement model with serially correlated unobserved state variables from the previous subsection. I briefly outline a way to solve the model by approximating the expected value function, but defer the details to Appendix A.1. I then motivate and develop a recursive method to integrate out the serially correlated errors in the computation of the marginal likelihood function; strategies for its maximization and the simultaneous solution of the model are again deferred to Appendix A.2.

2.2.1 The Expected Value Function

From (2) it is clear that in order to obtain the value function, I need to compute its conditional expectation. In fact, the computation of the likelihood function actually requires the expected value rather than the value itself (see Section 2.2.2). Thus, this section describes the steps necessary to numerically approximate the expected value as a function of all possible states:

$$EV_\theta(x,\varepsilon) = \sum_{x'} \int \max_{i\in\{0,1\}} \left\{u_\theta(i,x') + \varepsilon'(i) + \beta EV_\theta(x',\varepsilon')\right\} p_\varepsilon(d\varepsilon'|\varepsilon;\theta) p_x(x'|x;\theta) \equiv T(EV_\theta)(x,\varepsilon) \tag{8}$$

Keeping the original time structure of the expectation (6) in mind, the expectation on the leftmost side of (8) is—strictly speaking—taken at time t, while the one on the right hand side (within the max operator) is taken at time $t+1$. But since the value function and its expectation are time invariant, given state (x,ε), the same unknown function EV_θ appears on both the left and the right sides of the equation. Therefore, EV_θ is the solution to the functional equation

$$EV_\theta(x,\varepsilon) = T(EV_\theta)(x,\varepsilon) \tag{9}$$

and thus a fixed point of the non-linear operator T. Moreover, since T can be shown to have the contraction mapping property (Rust, 1988), this fixed point is unique and attractive.

The numerical approximation of (8) involves three main computational tasks:[5] First, I need to approximate the integral in (8) by numerical quadrature. Second, I have to approximate the continuous function EV_θ by a finite number of parameters, for example by interpolation. Finally, since EV_θ is only defined implicitly as a fixed point of T—and I therefore cannot evaluate it directly—I need to solve for the parameters of the function approximation by solving a non-linear system (or fixed point iteration).

Since the approximation of the expected value function is not the main subject of this paper, and, moreover, since the proposed method to integrate the likelihood function is independent of its maximization as well as the approximation methods for the EV function to the extend discussed in Section 3.4, I defer the precise description to Appendix A.1.

[5] Generally, there is one more task necessary, namely maximizing the utility and continuation value, in order to obtain the current value as a function of the states. However, since the choice set is discrete and unordered, the maximization must be done by complete enumeration.

2.2.2 The Likelihood Function

In this subsection, I derive the (marginal) likelihood function for the bus engine replacement model with serially correlated unobserved state variables, and formulate it such that the dimensionality of the numerical integration only depends on the number of choices N, and not on the time horizon of the observation, T. In a second step, I sketch a numerical procedure to solve this formulation by *recursive likelihood function integration* (RLI) to high accuracy, using standard deterministic quadrature and interpolation rules.[6] The numerical properties of the RLI method and its applicability to the estimation of general dynamic Markov models with unobserved serially correlated states are formally derived in Section 3.

The *marginal* likelihood function of observing a particular history of state transition and maintenance decisions for one individual bus derives as follows:

$$L_T(\theta) \equiv P_{xi}(\{x_t, i_t\}_{t=1}^T | \{x_0, i_0\}; \theta) \tag{10}$$

$$= \int \cdots \int p_\varepsilon(\varepsilon_0; \theta) P_{xi\varepsilon}(\{x_t, i_t, \varepsilon_t\}_{t=1}^T | \{x_0, i_0, \varepsilon_0\}; \theta) d\varepsilon_0 d\varepsilon_1 \dots d\varepsilon_T \tag{11}$$

where the integration limits are ignored for better readability.

The likelihood function of the full panel computes as the product of the likelihood functions of the individual buses, since the state variables are assumed to be independently distributed across buses. Incorporating the assumption that all state transitions are Markov, I can factorize the likelihood of observing a particular time series as

$$P_{xi\varepsilon}(\{x_t, i_t, \varepsilon_t\}_{t=1}^T | \{x_0, i_0, \varepsilon_0\}; \theta) = \prod_{t=2}^{T} p_{xi\varepsilon}(x_t, i_t, \varepsilon_t | x_{t-1}, i_{t-1}, \varepsilon_{t-1}; \theta) \tag{12}$$

I can further decompose the joint transition probability density in (12), using the fact that, given x_t and ε_t, i_t is independent of i_{t-1}, ε_{t-1}, and x_{t-1}, as well as incorporating assumption (5):

$$p_{xi\varepsilon}(x_t, i_t, \varepsilon_t | x_{t-1}, i_{t-1}, \varepsilon_{t-1}; \theta) = p_{i|x\varepsilon}(i_t | x_t, \varepsilon_t; \theta) p_\varepsilon(\varepsilon_t | i_{t-1}, \varepsilon_{t-1}; \theta) p_x(x_t | x_{t-1}, i_{t-1}; \theta) \tag{13}$$

For notational simplicity, I define

$$m_{it} \equiv u_\theta(i, x_t) + \beta \mathbb{E}[V_\theta(x_{t+1}, \varepsilon_{t+1}) | i, x_t, \varepsilon_t; \theta]. \tag{14}$$

While $p_\varepsilon(\varepsilon_t | i_{t-1}, \varepsilon_{t-1}, \theta)$ is determined by (7) and $p_x(x_t | x_{t-1}, i_{t-1})$ is estimated independently (and therefore omitted from now on),[7] the density function of the conditional decision proba-

[6] This is not to be confused with the recursive maximum likelihood estimation (RMLE) algorithm of Kay (1983) for the estimation of AR processes, which allows one to recursively update maximum likelihood estimates to higher order AR models.

[7] Since one can estimate the mileage transition probabilities separately, they only add a multiplicative constant to the likelihood function of $\theta = \{\theta_1, RC, \rho\}$. Thus, I omit the corresponding term of the likelihood function (and one should do so in the actual maximization for scaling reasons).

bility, $p_{i|x\varepsilon}(i_t|x_t, \varepsilon_t, \theta)$, is given by

$$p_{i|x\varepsilon}(1|x_t, \varepsilon_t(0), \varepsilon_t(1); \theta) = \mathbb{1}(m_{1t} + \varepsilon_t(1) > m_{0t} + \varepsilon_t(0)) \tag{15}$$

$$p_{i|x\varepsilon}(0|x_t, \varepsilon_t(0), \varepsilon_t(1); \theta) = \mathbb{1}(m_{1t} + \varepsilon_t(1) \leq m_{0t} + \varepsilon_t(0)) \tag{16}$$

where $\mathbb{1}(\cdot)$ is the index function that is equal to one if its argument is true, and zero otherwise; note that the conditional decision probabilities are actually degenerate, because—loosely speaking—there is no randomness left, given ε_t.

Finally, exploiting the Markov structure for the integration, and dropping parameter dependence for better readability, I can write the likelihood function (11) as

$$\begin{aligned} L_T(\theta) = \int \cdots \int p_\varepsilon(\varepsilon_0; \theta) \prod_{t=1}^{T-1} p_{i|x\varepsilon}(i_t|x_t, \varepsilon_t; \theta) p_\varepsilon(\varepsilon_t|i_{t-1}, \varepsilon_{t-1}; \theta) \\ \cdot \left(\int p_{i|x\varepsilon}(i_T|x_T, \varepsilon_T; \theta) p_\varepsilon(\varepsilon_T|i_{T-1}, \varepsilon_{T-1}; \theta) d\varepsilon_T \right) d\varepsilon_0 \dots d\varepsilon_{T-1} \end{aligned} \tag{17}$$

To numerically approximate (17), I define the following recurrence relation:

$$g_t(\varepsilon) = \begin{cases} 1 & t > T \\ \int p_{i|x\varepsilon}(i_t|x_t, \varepsilon'; \theta) p_\varepsilon(\varepsilon'|i_{t-1}, \varepsilon; \theta) g_{t+1}(\varepsilon') d\varepsilon' & \text{otherwise} \end{cases} \tag{18}$$

Now, given $g_{t+1}(\cdot)$, I can numerically approximate the function $g_t(\cdot)$ using both numerical integration and function approximation. Since $g_t(\cdot)$ is known to be unity for $t > T$, I can use backward iteration starting from $g_T(\cdot)$ to solve for $g_0(\cdot)$, which is the approximation of the likelihood function $L_T(\theta)$. Note that this procedure is analogous to solving for the value function of a finite horizon, discrete time dynamic programming problem by backward iteration. Algorithm 1 proposes a simple implementation of the procedure.[8]

Algorithm 1 Computation of the likelihood function (17) by recursive likelihood function integration (RLI).

```
Γ ← initialize grid over support of ε with D elements
ĝ(·) ← initialize interpolant with nodes {(e, g̃_e)}_{e∈Γ} to unity
for t = T, ..., 1 do
    for e ∈ Γ do
        g̃_e ← approximate ∫ p_{i|xε}(i_t|x_t, ε'; θ) p_ε(ε'|i_{t-1}, ε; θ) ĝ(ε') dε'
    end for
    ĝ(·) ← construct interpolant with nodes {(e, g̃_e)}_{e∈Γ}
end for
L_Tθ ← ∫ p_ε(ε_0; θ) ĝ(ε') dε'
return L_Tθ
```

Note that each integral over ε_t is generally still N-dimensional. Thus, the procedure decomposes the $T \cdot N$-dimensional integral of (11) to an N-dimensional integration that is repeated

[8] Algorithm 1 is generic with respect to both the numerical integration scheme and the function approximation schemes, as long as the latter depend on function evaluations only. In particular, formulating the function approximation using callback functions allows for fully flexible and adaptive interpolation grids, if desired.

$D \cdot T$ times, where D is the number of nodes used for the approximation of $g_t(\cdot)$. Since the computational complexity of deterministic numerical integration is generally exponential in the number of dimensions, this reduction is highly desirable even for large D, because it enters the complexity of the overall algorithm linearly[9]

$$O(\exp(T \cdot N)) \gg O(D \cdot T \exp(N)) \tag{19}$$

Given that serial correlation is only allowed in some dimensions, but not all, I can potentially replace parts of the integral in (18) by a closed form solution; this is particularly the case if the cumulative distribution of those unobserved state variables that are not serially correlated does have a closed form. Recall that the integration over ε_t is really N-dimensional, thus 2-dimensional in the model under consideration:

$$\iint p_{\varepsilon(0)}(\varepsilon_t(0)|i_{t-1}, \varepsilon_{t-1}(0); \theta) p_{\varepsilon(1)}(\varepsilon_t(1); \theta) p_{i|x\varepsilon}(i_t|x_t, \varepsilon_t(0), \varepsilon_t(1); \theta) d\varepsilon_t(1) d\varepsilon_t(0) \tag{20}$$

Using (15), I can write the integral over $\varepsilon_t(1)$ in terms of its cumulative distribution function $F_{\varepsilon(1)}(\cdot; \theta)$,

$$\begin{aligned}
&\int_{-\infty}^{\infty} \mathbb{1}(\varepsilon_t(1) > m_{0t} - m_{1t} + \varepsilon_t(0)) p_{\varepsilon(1)}(\varepsilon_t(1); \theta) d\varepsilon_t(1) && (21)\\
&\quad = \int_{m_{0t} - m_{1t} + \varepsilon_t(0)}^{\infty} p_{\varepsilon(1)}(\varepsilon_t(1); \theta) d\varepsilon_t(1) && (22)\\
&\quad = 1 - F_{\varepsilon(1)}(m_{0t} - m_{1t} + \varepsilon_t(0); \theta) && (23)
\end{aligned}$$

which no longer involves numerical quadrature if an analytical formula for $F_{\varepsilon(1)}$ exists.

The actual maximization of the likelihood function and the simultaneous solution of the EV problem (8) is deferred to Appendix A.2.

3 Recursive Likelihood Function Integration

This section derives the numerical properties of the recursive likelihood function integration method (RLI) as outlined in the previous section, including error analysis, convergence rates, and the necessary properties of the model for the theoretical results to be applicable. The section is mostly self-contained and structured very strictly in order to allow cross-referencing.

The section is structured as follows: First, to obtain a unified nomenclature, the concepts of numerical quadrature and interpolation are introduced (Subsections 3.1 and 3.2, respectively); moreover, particular methods are briefly presented and analyzed for their applicability in the RLI context. As a result, the convergence speed for parametric integration is derived in dependence of the convergence speed of the quadrature and interpolation methods under appropriate assumptions. Second, a recursive version of parametric integration is defined and analyzed for

[9] In this context, the $O(f(y))$ notation for the computational complexity of an algorithm reads as follows: There exists a constant $K > 0$ such that the number of iterations needed for an algorithm to complete a task of size y is bounded by $K \cdot f(y)$.

error propagation (Subsection 3.3); again, also the convergence speed of the numerical approximation by recursively applying quadrature and interpolation is derived in dependence of the convergence speed of the quadrature and interpolation methods under appropriate assumptions. The subsection is concluded with several analytical and numerical examples that compare the method to Monte Carlo integration, and confirm the theoretically derived error and convergence behavior. Finally, the scope of applicability of the method to integrate out serially correlated unobserved state variables to obtain the marginal likelihood function is analyzed (Subsection 3.4).

As a result, I find that the RLI method turns out to feature very desirable convergence and error properties by largely preserving the convergence properties of the underlying quadrature and interpolation methods, and that it is applicable to a wide range of dynamic Markov models with serially dependent unobserved states.

3.1 Numerical Quadrature

Definition 1 (Kernel Integral)**.** Given a function $f : \mathbb{R}^m \supseteq D \to \mathbb{R}$, the integral of f against a non-negative and bounded *kernel* or *weighting function* $q : \mathbb{R}^m \supseteq D \to [0, a]$ with $a \in \mathbb{R}_+$ is denoted by

$$I_f = \int_D f(x)q(x)dx \tag{24}$$

where $\int_D q(x)dx \leq 1$; note that only finite intervals are considered, i.e. $I_f < \infty$.

Remark 1 (Role of Kernel)**.** Note that the multiplication of the integrand f by a weighting function q in Definition 1 is w.l.o.g., because the kernel can be chosen as unity. However, it is generally needed to cover integrals over unbounded domains (e.g. $D = \mathbb{R}^m$).

Definition 2 (Approximation by Quadrature)**.** $\hat{I}_f$ is an *approximation* of I_f by (numerical) quadrature if

$$I_f = \hat{I}_f + \epsilon_f^Q, \quad |\epsilon_f^Q| \ll 1 \tag{25}$$

where ϵ_f^Q is the *approximation error*.

In the context of this paper, I limit my attention to numerical quadrature rules that comply with the following definition:

Definition 3 (Quadrature Rule)**.** A *quadrature rule* is any systematic choice of nodes and weights $\{(x_i, \omega_i)\}_{i=1}^{n^Q}$ such that I_f is approximated by

$$\hat{I}_f = \sum_{i=1}^{n_Q} \omega_i f(x_i) \tag{26}$$

and where $\{(x_i, \omega_i)\}_{i=0}^{n^Q}$ depend deterministically on n^Q, but not on f.

Example 1 (Compound Simpson Integration in One Dimension)**.** The rule of Simpson approximates the integrand f (or, more precisely, $f \cdot q$) by a quadratic polynomial, which can be integrated analytically; while the quadratic approximation is often accurate enough *locally*, but not over the whole domain, it is common practice to sub-divide the domain and compute the

integral over each subinterval using the Simpson rule locally.[10] Therefore, the *compound* (or *composite*) Simpson rule with n_Q (even) subintervals on $[a, b]$ reads as

$$\hat{I}_f = \frac{b-a}{3n_Q} \sum_{i=1}^{n_Q/2} f(x_{2j-2}) + 4f(x_{2j-1}) + f(x_{2j}) \tag{27}$$

where $x_0 = a$ and $x_{n_Q} = b$; see, for example, Davis and Rabinowitz (1984).

Example 2 (Gaussian Quadrature in One Dimension)**.** The n_Q-node Gaussian quadrature rule in one dimension implements (26) by choosing the integration nodes x_i as the roots of the degree n_Q polynomial of the family of polynomials that are mutually orthogonal with respect to the weighting function q.[11] The corresponding weights ω_i are chosen such that every polynomial of degree $2_{n_Q} - 1$ is integrated *exactly*; see, for example, Davis and Rabinowitz (1984).

Remark 2 (Non-Constant Rules)**.** Note that Definition 3 rules out two popular approaches to numerical integration, namely Monte Carlo integration (non-deterministic) and adaptive integration methods (dependence on integrand).

Definition 4 (Convergence of Quadrature Rule)**.** Given $f \in C^i$, the quadrature rule converges at rate s_Q, if

$$|\epsilon_f^Q| = O(n_Q^{-s_Q}) \quad \Leftrightarrow \quad \forall f \in C^i : \exists k < \infty : |\epsilon_f^Q| \leq k n_Q^{-s_Q} \tag{28}$$

Note that by C^i, I refer to the space of functions that are i times continuously differentiable, and for which the ith derivative is, moreover, bounded. Since functions in this context can be multivariate, this includes all partial derivatives, and must hold for all dimensions.

Remark 3 (Multivariate Integrals)**.** Note that since the integral (24) can be multivariate, the convergence rate as defined in Definition 4 is the total rate over *all* dimensions.

Example 1 (Compound Simpson Integration in One Dimension, continued)**.** Given that $f \in C^4$, the compound Simpson rule converges at rate 4: $|\epsilon_f^Q| = O(n_Q^{-4})$ (Davis and Rabinowitz, 1984).

Example 2 (Gaussian Quadrature in One Dimension, continued)**.** The convergence of Gaussian quadrature rules is exponential for sufficiently smooth integrands:

$$f \in C^{2n_Q} \;\Rightarrow\; |\epsilon_f^Q| = O(r^{-2n_Q}) \tag{29}$$

for some $r > 1$; see e.g. Davis and Rabinowitz (1984).

Remark 4 (Convergence of Monte Carlo Integration)**.** Note that while Monte Carlo integration is said to converge at rate 1/2, it does not do so in the sense of Definition 4, because it only converges in a mean square sense, rather than in maximum absolute error terms, because only

[10] The Simpson rule corresponds to the 3-point version of the Newton-Cotes (NC) rules, which generally approximate the integrand by a degree $n_Q - 1$ polynomial, which is then integrated analytically. However, due to changing signs of the (potentially large valued) coefficients, NC rules of higher order suffer from large roundoff errors on finite precision architectures. Therefore, approximating integrals by sub-dividing the domain and computing the integral over each subinterval using a lower order NC formula is a widely used practice.

[11] A family of polynomials $\{\varphi_k(y)\}_{k=0}^{\infty}$ with inner product $\langle \varphi_k, \varphi_l \rangle = \int_a^b \varphi_k(y)\varphi_l(y)q(y)dy$ is orthogonal with respect to weighting function q on $[a, b] \subseteq \mathbb{R}$ if $\langle \varphi_k, \varphi_l \rangle = 0 \;\forall k, l : k \neq l$. Given a weighting function as in Definition 1 which is, moreover, square integrable, a family of orthogonal polynomials can always be constructed using the Gram-Schmidt procedure; however, numerically more favorable methods exist (see, for example Press et al., 2007). A family of *monic* polynomials (leading coefficient equals 1) that is orthogonal w.r.t. a particular weighting function is, moreover, unique.

the *standard deviation* of the approximation of I_f decays at rate 1/2, but nothing can be said about the error of the actual approximation.

Definition 5 (Parametric Kernel Integral)**.** Given a function $f : \mathbb{R}^{m_x} \times \mathbb{R}^{m_y} \supseteq D \times E \to \mathbb{R}$, the *parametric* kernel integral of f is denoted by the function

$$I_f(y) = \int_D f(x, y)q(x)dx \tag{30}$$

where $I_f : \mathbb{R}^{m_y} \to \mathbb{R}$. Its approximation $\hat{I}_f(y)$ is given by

$$I_f(y) = \hat{I}_f(y) + \epsilon_f^Q(y) \tag{31}$$

where $\epsilon_f^Q(y)$ is the approximation error in dependence of the parameter y.

Definition 6 (Parametric Form)**.** Given a function $f : \mathbb{R}^{m_x} \times \mathbb{R}^{m_y} \supseteq D \times E \to \mathbb{R}$, $f_{\bar{y}}(x) \equiv f(x, \bar{y}) = f(x, y)\big|_{y=\bar{y}}$ and $f_{\bar{x}}(y) \equiv f(\bar{x}, y) = f(x, y)\big|_{x=\bar{x}}$ denote the *parametric form* of f with respect to y and x, respectively.

Loosely speaking, the parametric form of f fixes one of its two (potentially multivariate) arguments.

Definition 7 (Point-wise and Uniform Convergence of Function Sequences)**.** Consider a sequence of functions with domain D, $f_n : D \to \mathbb{R}$, $n \in \mathbb{N}$. This sequence is said to converge *point-wise* to f, iff

$$\forall x \in D, \epsilon > 0 : \exists N \in \mathbb{N} : \forall n > N : |f(x) - f_n(x)| < \epsilon \tag{32}$$

or, equivalently,

$$\forall x \in D : \lim_{n\to\infty} |f(x) - f_n(x)| = 0 \tag{33}$$

The sequence is said to converge *uniformly*, iff

$$\forall \epsilon > 0 : \exists N \in \mathbb{N} : \forall n > N, x \in D : |f(x) - f_n(x)| < \epsilon \tag{34}$$

where N is independent of x (in contrast to point-wise convergence), or, equivalently,

$$\lim_{n\to\infty} \sup_{x\in D} |f(x) - f_n(x)| = 0. \tag{35}$$

Note that uniform convergence implies point-wise convergence, but not vice-versa.

The sequence is said to converge (uniformly) *at rate* s, iff

$$\exists k < \infty : \forall x \in D : |f(x) - f_n(x)| \leq kn^{-s} = O(n^{-s}). \tag{36}$$

Definition 8 (Uniform Convergence of Parametric Integration)**.** A quadrature rule for parametric integration as in Definition 5 converges uniformly, if for all integrands $f \in C^i$,

$$\lim_{n_Q\to\infty} \sup_{y\in E} |\epsilon_f^Q(y)| = 0. \tag{37}$$

Moreover, it converges at rate s_Q, if

$$\|\epsilon_f^Q\|_\infty \equiv \sup_{y\in E} |\epsilon_f^Q(y)| = O(n_Q^{-s_Q}). \tag{38}$$

Example 1 (Compound Simpson Integration in One Dimension, continued)**.** The error of the compound Simpson rule applied to the parametric integral (30) with $f \in C^4$ reads as

$$\forall y : \exists \xi \in (a,b) : \epsilon_f^Q(y) = \frac{(b-a)^5}{180 n_Q^{-4}} \frac{\partial^4}{\partial x^4} f(\xi, y) \tag{39}$$

Therefore, point-wise convergence is obvious:

$$\forall y : \exists k < \infty : |\epsilon_f^Q(y)| \leq k n_Q^{-4} \tag{40}$$

Moreover, since the definition of C^i requires the i derivative to be continuous and bounded also in y, its maximum w.r.t. y is finite and attained in E, if E is compact. Therefore, convergence is uniform:

$$\|\epsilon_f^Q\|_\infty = \frac{(b-a)^5}{180 n_Q^{-4}} \max_y \max_x \left| \frac{\partial^4}{\partial x^4} f(x,y) \right| \tag{41}$$

implying

$$\exists k < \infty : \forall y : \|\epsilon_f^Q\|_\infty \leq k n_Q^{-4}. \tag{42}$$

Example 2 (Gaussian Quadrature in One Dimension, continued)**.** The derivation of uniform convergence for the approximation of the parametric integral using Gaussian quadrature is analogous to Example 1.

Remark 5 (Preservation of Smoothness)**.** Note that the approximation of $I_f(y)$ by $\hat{I}_f(y)$ through a quadrature rule as defined by Definition 3 preserves smoothness in y, since $\hat{I}_f(y)$ is just a weighted sum of functions in y:

$$\hat{I}_f(y) = \sum_{i=1}^{n_Q} \omega_i f_{\bar{x}_i}(y) \tag{43}$$

Therefore, $f \in C^i \;\Rightarrow\; \hat{I}_f \in C^i$. However, the smoothness is not necessarily preserved by other numerical integration methods, such as adaptive quadrature or Monte Carlo integration, because the x_i either depend (non-smoothly) on f, or are non-deterministic; also note that if only finite integrals and convergent quadrature methods are considered, the potentially infinite sum (as $n_Q \to \infty$) is always bounded.

3.2 Interpolation

Definition 9 (Interpolation)**.** Given a function $f : \mathbb{R}^m \supseteq E \to \mathbb{R}$, n_I pairs of argument values and the corresponding function values, $\{(y_i, f(y_i))\}_{i=1}^{n_I}$, and an Ansatz-function $\phi(y; \mathbf{a})$ with n_I parameters $\mathbf{a} = (a_1, \ldots, a_{n_I})$, $\mathcal{I}_f \equiv \phi(\cdot; \mathbf{a})$ is called the interpolant of f, if the parameters $\mathbf{a}$ are chosen such that

$$\phi(y_i; \mathbf{a}) = f(y_i) \; \forall i \tag{44}$$

The corresponding interpolation error $\epsilon_f^I(y)$ is defined by

$$f(y) = \mathcal{I}_f(y) + \epsilon_f^I(y) \tag{45}$$

and is 0 at the interpolation nodes, $\epsilon_f^I(y_i) = 0$.

Remark 6 (Alternative Interpolation Definitions)**.** While many popular interpolation schemes can be defined in terms of their Ansatz-function as in Definition 9—which is itself often a sum of basis functions, such as the different families of orthogonal polynomials or the B-splines—, alternative notations are sometimes more intuitive and closer to their implementation; see e.g. the piecewise linear interpolation in Example 4.

Example 3 (Chebyshev Polynomials)**.** Given a set of n_I interpolation nodes y_i, the n_I coefficients of an order $n_I - 1$ polynomial can be computed such that the interpolation property (44) holds. However, since (i) the choice of the interpolation nodes is critical for convergence, and since (ii) directly solving for the coefficients can cause numerical problems, a popular choice for approximating f on $[0, 1]$ is Chebyshev interpolation:

$$\mathcal{I}_f(y) = \sum_{i=0}^{n_I-1} c_i T_i(y) \tag{46}$$

where

$$T_i(y) = 2yT_{i-1}(y) - T_{i-2}(y) \quad i = 2, \ldots, n_I - 1 \tag{47}$$

$$c_i = \frac{1 + \mathbb{1}(i > 0)}{n_Q} \sum_{j=0}^{n_I-1} f(y_{j+1}) T_i(y_{i+1}) \quad i = 0, \ldots, n_I - 1 \tag{48}$$

$$y_i = \cos\left(\pi \frac{(2i-1)}{n_I}\right) \quad i = 1, \ldots, n_I \tag{49}$$

and $T_0(y) = 1$ and $T_1(y) = y$; see, for example, Trefethen (2013).

Example 4 (Piecewise Linear Interpolation in One Dimension)**.** Piecewise linear interpolation on a grid $\{y_i\}_{i=1}^{n_I}$ is probably the simplest form of interpolation:

$$\mathcal{I}_{f,i}(y) = \theta f(y_i) + (1 - \theta) f(y_{i+1}), \quad \theta = 1 - \frac{y - y_i}{y_{i+1} - y_i} \tag{50}$$

where $i = \arg\max_i \{y_i : y_i \leq y\}$. Note that it is both a special case of piecewise polynomial as well as spline interpolation (see below).

Example 5 (Cubic Spline Interpolation in One Dimension)**.** Similar to piecewise linear interpolation, cubic spline interpolation constructs an interpolant by connecting several low-degree polynomials; since the resulting number of degrees of freedom is larger than the number of interpolation nodes, the remaining parameters are chosen such that the first and the second derivative of the interpolant are matched at the interpolation nodes by solving a linear system

of equations. Therefore, cubic spline interpolation reads as:

$$\mathcal{I}_{f,i}(y) = \sum_{j=0}^{3} a_{i,j} y^j \tag{51}$$

where $i = \arg\max_i \{y_i : y_i \leq y\}$, and the $a_{i,j}$ are chosen to solve the following linear system of equations:

$$f(y_i) = \mathcal{I}_{f,i}(y_i), \quad i = 1, \ldots, n_I \tag{52}$$

$$f(y_{i+1}) = \mathcal{I}_{f,i}(y_{i+1}), \quad i = 1, \ldots, n_I - 1 \tag{53}$$

$$\mathcal{I}^{(h)}_{f,i}(y_{i+1}) = \mathcal{I}^{(h)}_{f,i+1}(y_{i+1}), \quad i = 1, \ldots, n_I - 2,\ h = 1, 2 \tag{54}$$

Additionally, two remaining degrees of freedom need to be identified through a boundary condition. For a complete description of splines, see, for example, Kress (1998).

Definition 10 (Uniform Convergence for Interpolation)**.** An interpolation scheme converges uniformly, if for all functions $f \in C^i$,

$$\lim_{n_I \to \infty} \sup_{y \in E} |\epsilon^I_f(y)| = 0. \tag{55}$$

Moreover, it converges at rate s_I, if

$$\|\epsilon^I_f\|_\infty \equiv \sup_{y \in E} |\epsilon^I_f(y)| = O(n_I^{-s_I}). \tag{56}$$

Remark 7 (Multivariate Interpolants)**.** Note that since the interpolant in (44) can be multivariate, the convergence rate as defined in Definition 10 is the total rate over *all* dimensions.

Example 3 (Chebyshev Polynomials, continued)**.** Given a function $f \in C^i$, Chebyshev interpolation converges at rate i, given $n_I \geq i$: $\|\epsilon^I_f\|_\infty = O(n_I^{-i})$. If f is moreover analytic, convergence is exponential: $\|\epsilon^I_f\|_\infty = O(r^{-n})$ for some $r > 1$ (Trefethen, 2013).

Example 4 (Piecewise Linear Interpolation in One Dimension, continued)**.** Given a function $f \in C^2$, piecewise linear interpolation converges at rate 2; moreover, every continuous and bounded function over a compact interval is a uniform limit of piecewise linear continuous functions; see e.g. Kress (1998).

Example 5 (Cubic Spline Interpolation in One Dimension, continued)**.** Given a function $f \in C^4$, cubic spline interpolation converges at rate 4 (Kress, 1998).

Definition 11 (Preservation of Smoothness)**.** The interpolation scheme preserves smoothness up to order j, if:

$$f \in C^i \ \Rightarrow\ \mathcal{I}_f \in C^j \tag{57}$$

It fully preserves smoothness if, moreover, $i = j$.

Example 3 (Chebyshev Polynomials, continued)**.** Since all polynomials are C^∞, smoothness is preserved.

Example 4 (Piecewise Linear Interpolation in One Dimension, continued)**.** Since piecewise

linear interpolants are C^0 but require $f \in C^2$ to exhibit convergence rate 2, only continuity is preserved.

Example 5 (Cubic Spline Interpolation in One Dimension, continued)**.** Since cubic spline interpolants are C^2, but require $f \in C^4$ to exhibit convergence rate 4, smoothness is only preserved up to order 2.

Definition 12 (Approximation of Parametric Integration)**.** Consider a parametric integration problem as in Definition 5; its approximation by quadrature and interpolation is defined as

$$I_f(y) = \hat{I}_f(y) + \epsilon_f^Q(y) \tag{58}$$

$$= \mathcal{I}_{\hat{I}_f}(y) + \underbrace{\epsilon_{\hat{I}_f}^I(y) + \epsilon_f^Q(y)}_{\equiv \epsilon_{I_f}(y)} \tag{59}$$

where $\epsilon_{I_f}(y)$ denotes the overall approximation error.

Remark 8. Note that while ϵ_f^Q depends on n_Q only, $\epsilon_{\hat{I}_f}^I$ depends on n_I, but also on n_Q in a potentially non-monotone way through $\hat{I}_f$; consequently, the convergence rate results below will have to account for function sequences in multiple dimensions. Therefore, I will use the following equivalent notation for the approximation error(s) in proofs:

$$\epsilon_{I_f}(y) \equiv \epsilon_{n^Q}^Q(y) + \epsilon_{n^Q,n^I}^I(y) \tag{60}$$

Note that this can be understood as parametric forms of ϵ^Q w.r.t. n_Q and ϵ^I w.r.t. n_Q and n_I, respectively, and thus as n_Q and n_I being (discrete) function arguments rather than establishing function sequences, which is how I interpret it in the convergence proofs below, to stay consistent with Definition 7.

Proposition 1 (Convergence of Parametric Integration)**.** *Consider the approximation of a parametric integration problem by quadrature and interpolation as in Definition 12, where $f \in C^i$, and quadrature and interpolation methods that converge uniformly in the sense of Definitions 8 and 10, respectively. Then, the approximation of the parametric integration problem converges uniformly as n_Q and n_I tend to infinity:*

$$\lim_{n_Q, n_I \to \infty} \|\epsilon_{I_f}\|_\infty = 0. \tag{61}$$

Moreover, if the quadrature and interpolation methods converge at rates s_Q and s_I, respectively, and if the number of quadrature and interpolation nodes are chosen as $n_Q = n^\theta$ and $n_I = n^{(1-\theta)}$ with $\theta \in (0,1)$, then the approximation of the parametric integration problem in terms of total integrand evaluations, $n = n_Q n_I$, converges uniformly at rate $s = \min\{s_Q\theta, s_I(1-\theta)\}$:

$$\|\epsilon_{I_f}\|_\infty = O\left(n^{-\min\{s_Q\theta, s_I(1-\theta)\}}\right). \tag{62}$$

Proof. Since the overall approximation error is the sum of the respective quadrature and inter-

polation errors, the triangle inequality holds:

$$\|\epsilon_{I_f}\|_\infty \equiv \|\epsilon_f^Q + \epsilon_{\hat{I}_f}^I\|_\infty \equiv \|\epsilon_{n^Q}^Q + \epsilon_{n^Q,n^I}^I\|_\infty \tag{63}$$

$$\leq \|\epsilon_{n^Q}^Q\|_\infty + \|\epsilon_{n^Q,n^I}^I\|_\infty \tag{64}$$

From Definition 8, uniform convergence of the parametric integration is assured, i.e.

$$\lim_{n_Q\to\infty} \|\epsilon_{n^Q}^Q\|_\infty = 0, \tag{65}$$

and, if moreover the convergence is polynomial at a known rate,

$$\exists k < \infty : \|\epsilon_{n^Q}^Q\|_\infty \leq k n_Q^{-s_Q}. \tag{66}$$

Similarly, although $\epsilon_{\hat{I}_f}^I \equiv \epsilon_{n^Q,n^I}^I$ not only depends on n_I, but also on n_Q in a potentially non-monotone way through $\hat{I}_f$, a uniformly convergent interpolation method in line with Definition 10 assures that

$$\forall n_Q \in \mathbb{N} : \lim_{n_I\to\infty} \|\epsilon_{n^Q,n^I}^I\|_\infty = 0, \tag{67}$$

and, if moreover the convergence is polynomial at a known rate,

$$\forall n_Q \in \mathbb{N} : \exists k_{n_Q} < \infty : \|\epsilon_{n^Q,n^I}^I\|_\infty \leq k_{n_Q} n_I^{-s_I}. \tag{68}$$

Note that at this point, we have already proved point-wise convergence of the parametric integral, as n_Q and n_I jointly go to infinity (where the partial convergence of $\|\epsilon_{n^Q}^Q\|_\infty$ is already uniform).

To prove uniform convergence of $\|\epsilon_{n^Q,n^I}^I\|_\infty$, it is important to analyze the asymptotic behavior of the interpolation error with regard to n_Q: Given the corresponding assumptions about the choice of the quadrature and interpolation method as well as the parametric integrand, as n_Q approaches infinity, the quadrature error vanishes, and the (uniformly convergent) interpolation is carried out on the "true" parametric integral; formally, this implies convergence—besides for every finite n_Q—in particular for the limit:

$$\lim_{n_I\to\infty} \lim_{n_Q\to\infty} \|\epsilon_{n^Q,n^I}^I\|_\infty = 0, \tag{69}$$

and thus

$$\lim_{n_I\to\infty} \sup_{n_Q\in\mathbb{N}} \|\epsilon_{n^Q,n^I}^I\|_\infty = 0. \tag{70}$$

Therefore, uniform convergence of parametric integration follows from bounding the triangle inequality (64) by the limits (65) and (70), yielding (61).

With polynomial convergence rates of the quadrature and interpolation methods, an equivalent argument implies that

$$\lim_{n_Q\to\infty} k_{n_Q} < \infty. \tag{71}$$

Together, (68) and (71) imply a finite bound on k_{n_Q}:

$$k^* \equiv \sup_{n_Q \in \mathbb{N}} k_{n_Q} < \infty \tag{72}$$

and thus (68) can be further bounded by

$$\|\epsilon^I_{n^Q,n^I}\|_\infty \leq k^* n_I^{-s_I} \quad \forall n_Q \in \mathbb{N}. \tag{73}$$

Since the triangle inequality (64) can be further bounded by

$$\|\epsilon^Q_{n^Q}\|_\infty + \|\epsilon^I_{n^Q,n^I}\|_\infty \leq 2\max\{\|\epsilon^Q_{n^Q}\|_\infty, \|\epsilon^I_{n^Q,n^I}\|_\infty\}, \tag{74}$$

Definitions 8 and 10 ensure that there exists constants k and k^* such that

$$\begin{aligned} 2\max\{\|\epsilon^Q_{n^Q}\|_\infty, \|\epsilon^I_{n^Q,n^I}\|_\infty\} &\leq 2\max\{k n_Q^{-s_Q}, k^* n_I^{-s_I}\} && (75)\\ &\leq \tilde{k}\max\{n_Q^{-s_Q}, n_I^{-s_I}\} && (76)\\ &= \tilde{k}\max\{n^{-s_Q\theta}, n^{-s_I(1-\theta)}\} && (77)\\ &= \tilde{k} n^{-\min\{s_Q\theta, s_I(1-\theta)\}} && (78) \end{aligned}$$

where $\tilde{k} = 2\max\{k, k^*\}$. □

Remark 9 (Asymptotic Convergence Rate)**.** Note that Proposition 1 constitutes an "asymptotic" convergence rate, which might be observed only as n_Q—and therefore n—approaches infinity. However, in the numerical examples we present below, this point has no practical relevance, as the result manifests itself already for small n_Q.

Corollary 1 (Optimal Node Distribution)**.** *Given the assumptions of Proposition 1 and known convergence rates of the respective methods, optimal convergence of the parametric integration can be obtained by* minimizing *the error bound* (62) *with respect to* θ*:*

$$\frac{s_I}{s_Q + s_I} = \arg\min_\theta n^{-\min\{s_Q\theta, s_I(1-\theta)\}} \tag{79}$$

Example 6 (Optimal Node Distribution)**.** If $s_Q = s_I$, it is optimal to balance quadrature and interpolation nodes by choosing $n_Q = n_I = \sqrt{n}$.

Remark 10 (Convergence Rate in Limit Cases)**.** If the convergence rate of either quadrature or interpolation is very much higher than the other one, formally if $s_Q \ll s_I$ or $s_Q \gg s_I$, the optimal θ tends to 1 or 0, respectively. In the limiting case, however, the convergence rate of the recursive parametric integration turns out to be $s = \min\{s_Q, s_I\}$.

3.3 Recursive Parametric Integration

Definition 13 (Recursive Parametric Integral)**.** Given a function $f : \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{N} \supseteq D \times D \times \mathbb{N} \to \mathbb{R}$, and a kernel $q : \mathbb{R}^m \supseteq D \to [0, a], a \in \mathbb{R}_+$ in the sense of Definition 1, the *recursive parametric*

integral of f of order $T < \infty$ is denoted by the function

$$L_f^T = \int_{D^T} \cdots \int \prod_{t=1}^{T} f_t(x_t, x_{t-1}) q_t(x_t) dx_1, \ldots, dx_T \tag{80}$$

where f_t and q_t are parametric forms of f and q, respectively, in the sense of Definition 6: $f_t(x_t, x_{t-1}) \equiv f(x_t, x_{t-1}, t)$ and $q_t(x_t) \equiv q(x_t, t)$ with $\forall t : \int_D q(x_t, t) dx_t \leq 1$, and x_0 is given. Its approximation $\hat{L}_f^T$ is defined by

$$L_f^T = \hat{L}_f^T + \epsilon_{L_f^T} \tag{81}$$

where $\epsilon_{L_f^T} \ll 1$ is the approximation error.

Also see Remark 1 for the role of the kernel.

Proposition 2 (Convergence of Recursive Parametric Integration)**.** *Given a recursive parametric integral L_f^T as in Definition 13 with the* restricted *integrand $f : \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{N} \supseteq D \times D \times (1, \ldots, \bar{T}) \to [0, 1]$, and $f_t, q_t \in C^i, t = 1, \ldots, \bar{T}$, with T bounded by $T \leq \bar{T} < \infty$, consider its approximation by recursive application of the parametric integral approximation using quadrature and interpolation as in Definition 12, where the quadrature and the interpolation methods converge uniformly in the sense of Definitions 8 and 10, respectively, and where the interpolation method is sufficiently smoothness preserving as by Definition 11. Then,*

1. *for fixed T, recursive parametric integration converges, i.e. the maximum error of the approximation of the recursive integral, $\hat{L}_f^T$, vanishes as n_Q and n_I tend to infinity:*[12]

$$\lim_{n_Q, n_I \to \infty} |\epsilon_{L_f^T}| = 0. \tag{82}$$

2. *fixing the number of quadrature and interpolation nodes for each iteration to n_Q and n_I, respectively, the maximum approximation error of the recursive integral, as a function of T, is bounded linearly:*

$$|\epsilon_{L_f^T}| = O(T). \tag{83}$$

3. *if the quadrature and interpolation method converge at rates s_Q and s_I, respectively, the convergence rate of the overall approximation error $|\epsilon_{L_f^T}|$ in terms of total integrand evaluations, n, is given by*

$$|\epsilon_{L_f^T}| = O\left(T \left(\frac{n}{T}\right)^{-\min\{s_Q \theta, s_I (1-\theta)\}}\right) \tag{84}$$

where $n_{Q,t} = (n/T)^{\theta}$ and $n_{I,t} = (n/T)^{1-\theta}$ with $\theta \in (0, 1)$.

Proof. First, re-write the definition of the recursive parametric integral (80) to make the "last" dimension of the integral explicit, which can then be approximated using quadrature and inter-

[12] W.l.o.g., I assume that n_Q and n_I are the same in each iteration t to simplify notation.

polation as in Definition 12, yielding an explicit approximation error formulation:

$$L_f^T \equiv \int_{D^T} \cdots \int \prod_{t=1}^{T} f_t(x_t, x_{t-1}) q_t(x_t) dx_1, \ldots, dx_T \tag{85}$$

$$= \int_{D^{T-1}} \cdots \int \prod_{t=1}^{T-1} f_t(x_t, x_{t-1}) q_t(x_t) \Bigg(\int_D \underbrace{f_T(x_T, x_{T-1})}_{\equiv \bar{f}_T(x_T, x_{T-1})} q_t(x_T) dx_T \Bigg) dx_1, \ldots, dx_{T-1} \tag{86}$$

$$= \int_{D^{T-1}} \cdots \int \prod_{t=1}^{T-1} f_t(x_t, x_{t-1}) q_t(x_t) \left(\mathcal{I}_{\hat{I}_{\bar{f}_T}}(x_{T-1}) + \epsilon_{I_{\bar{f}_T}}(x_{T-1}) \right) dx_1, \ldots, dx_{T-1} \tag{87}$$

$$\begin{aligned} &= \int_{D^{T-2}} \cdots \int \prod_{t=1}^{T-2} f_t(x_t, x_{t-1}) q_t(x_t) \\ &\quad \cdot \Bigg(\int_D \underbrace{f_{T-1}(x_{T-1}, x_{T-2}) \mathcal{I}_{\hat{I}_{\bar{f}_T}}(x_{T-1})}_{\equiv \bar{f}_{T-1}(x_{T-1}, x_{T-2})} q_t(x_{T-1}) dx_{T-1} \Bigg) dx_1, \ldots, dx_{T-2} \\ &\quad + \underbrace{\int_{D^{T-1}} \cdots \int \prod_{t=1}^{T-1} f_t(x_t, x_{t-1}) q_t(x_t) \epsilon_{I_{\bar{f}_T}}(x_{T-1}) dx_1, \ldots, dx_{T-1}}_{\equiv \epsilon_T} \end{aligned} \tag{88}$$

$$\begin{aligned} &\equiv \int_{D^{T-2}} \cdots \int \prod_{t=1}^{T-2} f_t(x_t, x_{t-1}) q_t(x_t) \\ &\quad \cdot \Bigg(\int_D \bar{f}_{T-1}(x_{T-1}, x_{T-2}) q_t(x_{T-1}) dx_{T-1} \Bigg) dx_1, \ldots, dx_{T-2} + \epsilon_T \end{aligned} \tag{89}$$

Since the definition of the (recursive) parametric integrand in (88) is important, I restate it explicitly here:

$$\bar{f}_{t-1}(x_{t-1}, x_{t-2}) \equiv f_{t-1}(x_{t-1}, x_{t-2}) \mathcal{I}_{\hat{I}_{\bar{f}_t}}(x_{t-1}) \tag{90}$$

Since the interpolation method is chosen to be smoothness preserving (Definition 11), $f_T, f_{T-1} \in C^i \Rightarrow \bar{f}_{T-1} \in C^i$ (note that it is generally not bounded by 1 anymore).

Consequently, the approximation of the "last" dimension of the integral as above can be repeated recursively, yielding

$$\begin{aligned} L_f^T &= \int_{D^{T-s}} \cdots \int \prod_{t=1}^{T-s} f_t(x_t, x_{t-1}) q_t(x_t) \\ &\quad \cdot \Bigg(\int_D \bar{f}_{T-s+1}(x_{T-s+1}, x_{T-s}) q_{T-s+1}(x_{T-s+1}) dx_{T-s+1} \Bigg) dx_1, \ldots, dx_{T-2} + \sum_{t=T-s+2}^{T} \epsilon_t \end{aligned} \tag{91}$$

$$= \hat{I}_{\bar{f}_1} + \sum_{t=1}^{T} \epsilon_t \tag{92}$$

$$\equiv \hat{L}_f^T + \sum_{t=1}^{T} \epsilon_t \tag{93}$$

where

$$\epsilon_t \equiv \int_{D^{t-1}} \cdots \int \prod_{s=1}^{t-1} f_s(x_s, x_{s-1}) q_s(x_s) \epsilon_{I_{\bar{f}_t}}(x_{t-1}) dx_1, \ldots, dx_{t-1}, \quad t > 1 \tag{94}$$

and $\epsilon_1 \equiv \epsilon^I_{\bar{f}_1}$.

To obtain a worst case error estimate, note that

$$|\epsilon_t| \leq \int_{D^{t-1}} \cdots \int \prod_{s=1}^{t-1} f_s(x_s, x_{s-1}) q_s(x_s) |\epsilon_{I_{\bar{f}_t}}(x_{t-1})| dx_1, \ldots, dx_{t-1} \tag{95}$$

$$\leq \bar{\epsilon}_t \int_{D^{t-1}} \cdots \int \prod_{s=1}^{t-1} f_s(x_s, x_{s-1}) q_s(x_s) dx_1, \ldots, dx_{t-1} \tag{96}$$

$$\equiv \bar{\epsilon}_t L_f^{t-1} \tag{97}$$

where

$$\bar{\epsilon}_t \equiv \|\epsilon_{I_{\bar{f}_t}}\|_\infty \equiv \sup_{x_{t-1}} |\epsilon_{I_{\bar{f}_t}}(x_{t-1})| \tag{98}$$

and (95) follows from that fact that $f_t, q_t \geq 0$; note that in practice, sign changes in $\epsilon_{I_{\bar{f}_t}}$ will actually tend to cancel out the error, further decreasing it. Also, it is important to note that while $\bar{\epsilon}_t$ carries the recursion index t, it does per se not accumulate any error when iterating, as it is just the approximation error for a given function $I_{\bar{f}_t}$, which is indeed the result of the iteration, but per se does not "know" anything about its background (also see the proof of part 3 below).

I now show by induction that $\forall t \in \mathbb{N} : L_f^t \leq 1$:

$$L_f^1 = \int_D f_1(x_1, x_0) q_1(x_1) dx_1 \leq 1 \tag{99}$$

because $f_t, q_t \geq 0$, $f_t \leq 1$, and $\int_D q_t(x) dx \leq 1$, and

$$L_f^t \equiv \int_{D^{t-1}} \cdots \int \prod_{s=1}^{t-1} f_s(x_s, x_{s-1}) q_s(x_s) \Big(\int_D f_t(x_t, x_{t-1}) q_t(x_t) dx_t \Big) dx_1, \ldots, dx_{t-1} \tag{100}$$

$$\leq \int_{D^{t-1}} \cdots \int \prod_{s=1}^{t-1} f_s(x_s, x_{s-1}) q_s(x_s) \, 1 \, dx_1, \ldots, dx_{t-1} \tag{101}$$

$$\equiv L_f^{t-1} \tag{102}$$

for the same reason. Therefore, $\forall t \in \mathbb{N} : L_f^t \leq 1$, and thus (97) can further be bounded by

$$|\epsilon_t| \leq \bar{\epsilon}_t L_f^{t-1} \leq \bar{\epsilon}_t \tag{103}$$

Note that this step is critical for the proof, as it exploits the restriction of the integrand f to map to $[0, 1]$.

Recall that

$$L_f^T = \hat{L}_f^T + \sum_{t=1}^{T} \epsilon_t \tag{104}$$

From (103) is follows that

$$|L_f^T - \hat{L}_f^T| = \left| \sum_{t=1}^{T} \epsilon_t \right| \tag{105}$$

$$\leq \sum_{t=1}^{T} |\epsilon_t| \tag{106}$$

$$\leq \sum_{t=1}^{T} \bar{\epsilon}_t. \tag{107}$$

For the moment, assume that $\bar{\epsilon}_t$ is bounded for each t (which will be proved in the second part, c.f. Equations 118 and 124). Therefore, and since T is bounded by $\bar{T}$,

$$k \equiv \max_{t \in \{1,\dots,\bar{T}\}} \bar{\epsilon}_t, \tag{108}$$

exists, and fixing n_Q and n_I for all t allows to further bound (107) by

$$\sum_{t=1}^{T} \bar{\epsilon}_t \leq Tk, \tag{109}$$

which proves part 2 of the proposition (conditional on the assumption of bounded errors).

To prove convergence (parts 1 and 3), it is important to note that analogously to the potentially non-monotone dependence of the interpolation error from the number of quadrature nodes in the parametric integration problem (see Remark 8), also the maximum approximation error at iteration t, $\|\epsilon_{I_{\bar{f}_t}}\|_\infty$ depends on all previous pairs of numbers of quadrature and interpolation nodes, $(n_{Q,s}, n_{I,s})_{s=T}^{t+1}$, in a potentially non-monotone way, just through the definition of the integrand, $\bar{f}_t$ in Equation (90). To simplify notation, we assume (w.l.o.g) that the respective numbers of quadrature and interpolation nodes in all previous iterations was the same, namely n_Q and n_I, respectively.

Similarly to the proof of Proposition 1, the triangle inequality now reads as

$$\|\epsilon_{I_{\bar{f}_t}}\|_\infty \equiv \|\epsilon^Q_{\bar{f}_t} + \epsilon^I_{\hat{f}_{\bar{f}_t}}\|_\infty \equiv \|\epsilon^{Q,t}_{n_Q,n_I,\tilde{n}_Q} + \epsilon^{I,t}_{n_Q,n_I,\tilde{n}_Q,\tilde{n}_I}\|_\infty \tag{110}$$

$$\leq \|\epsilon^{Q,t}_{n_Q,n_I,\tilde{n}_Q}\|_\infty + \|\epsilon^{I,t}_{n_Q,n_I,\tilde{n}_Q,\tilde{n}_I}\|_\infty \tag{111}$$

where the errors now also depends on n_Q and n_I from previous iterations, and where the subscripts $\tilde{n}_Q$ and $\tilde{n}_I$ denote the dependence on the number of quadrature and interpolation nodes in the current iteration (as in the proof of Proposition 1). Consequently, the time index of the errors has no direct meaning for the convergence results at this point, but solely distinguished the different parametric integrands at the various iterations.

Given the assumptions of this proposition hold, it is clear that the parametric integrand (c.f. Equation 90) also satisfies the necessary smoothness (and its preservation to the degree that

uniform convergence is granted) as well as the boundedness conditions for this Proposition; note that no restrictions on the image space of the integrand are needed for this Proposition to hold, which in fact couldn't be guaranteed for the recursive integrand anyway. Therefore, for each choice of n_Q and n_I—no matter how long one has been iterating already, i.e. no matter how large $T - t$ is—point-wise convergence is granted for the quadrature error by Proposition 1:

$$\forall t \in \{1, \dots, \bar{T}\} : \forall (n_Q, n_I) \in \mathbb{N}^2 : \lim_{\tilde{n}_Q \to \infty} \|\epsilon^{Q,t}_{n_Q, n_I, \tilde{n}_Q}\|_\infty = 0 \tag{112}$$

and, if moreover the convergence is polynomial at known rates, and the interpolation method is fully smoothness preserving,

$$\forall t \in \{1, \dots, \bar{T}\} : \forall (n_Q, n_I) \in \mathbb{N}^2 : \exists k_{t,n_Q,n_I} < \infty : \|\epsilon^{Q,t}_{n_Q, n_I, \tilde{n}_Q}\|_\infty \leq k_{t,n_Q,n_I} \tilde{n}_Q^{-s_Q}. \tag{113}$$

(In the following, I will, for notational simplicity, skip the notion of $\forall t \in \{1, \dots, \bar{T}\}$ where adequate.)

Analogously to the proof of Proposition 1, the approximation of the parametric integrand converges to the true integrand as n_Q and n_I approach infinity, which implies (by the definition of compliant quadrature and interpolation methods)

$$\lim_{\tilde{n}_Q \to \infty} \lim_{n_Q, n_I \to \infty} \|\epsilon^{Q,t}_{n_Q, n_I, \tilde{n}_Q}\|_\infty = 0, \tag{114}$$

and thus

$$\lim_{\tilde{n}_Q \to \infty} \sup_{n_Q, n_I} \|\epsilon^{Q,t}_{n_Q, n_I, \tilde{n}_Q}\|_\infty = 0. \tag{115}$$

With known convergence rates, equivalently

$$\lim_{n_Q, n_I \to \infty} k_{t,n_Q,n_I} < \infty \tag{116}$$

and thus

$$k_t^* \equiv \sup_{(n_Q, n_I) \in \mathbb{N}^2} k_{t,n_Q,n_I} < \infty \tag{117}$$

Consequently, the convergence in (113) can actually be uniformly bounded:

$$\exists k_t^* < \infty : \forall (n_Q, n_I) \in \mathbb{N}^2 : \|\epsilon^{Q,t}_{n_Q, n_I, \tilde{n}_Q}\|_\infty \leq k_t^* \tilde{n}_Q^{-s_Q} \tag{118}$$

In particular, this holds for $n_Q = \tilde{n}_Q$:

$$\exists k_t^* < \infty : \forall n_I \in \mathbb{N} : \|\epsilon^{Q,t}_{n_Q, n_I}\|_\infty \leq k_t^* n_Q^{-s_Q} \tag{119}$$

For the interpolation error, the same argument that lead to (112) and (113) yields point-wise convergence of the interpolation error:

$$\forall (n_Q, n_I, \tilde{n}_Q) \in \mathbb{N}^3 : \lim_{\tilde{n}_I \to \infty} \|\epsilon^{I,t}_{n_Q, n_I, \tilde{n}_Q, \tilde{n}_I}\|_\infty = 0 \tag{120}$$

and, with known convergence rates,

$$\forall(n_Q, n_I, \tilde{n}_Q) \in \mathbb{N}^3 : \exists k_{t,n_Q,n_I,\tilde{n}_Q} < \infty : \|\epsilon^{I,t}_{n_Q,n_I,\tilde{n}_Q,\tilde{n}_I}\|_\infty \leq k_{t,n_Q,n_I,\tilde{n}_Q} \tilde{n}_I^{-s_I}. \tag{121}$$

Since—as above—

$$\lim_{\tilde{n}_I \to \infty} \sup_{n_Q,n_I,\tilde{n}_I} \|\epsilon^{I,t}_{n_Q,n_I,\tilde{n}_Q,\tilde{n}_I}\|_\infty = 0. \tag{122}$$

and, with known convergence rates,

$$k_t^{**} \equiv \sup_{(n_Q,n_I,\tilde{n}_Q)\in\mathbb{N}^3} k_{t,n_Q,n_I,\tilde{n}_Q} < \infty, \tag{123}$$

also the interpolation error can be bounded uniformly; moreover, with known convergence rates (together with directly equating $n_Q = \tilde{n}_Q$ and $n_I = \tilde{n}_I$):

$$\exists k_t^{**} < \infty : \forall n_Q \in \mathbb{N} : \|\epsilon^{I,t}_{n_Q,n_I}\|_\infty \leq k_t^{**} n_I^{-s_I} \tag{124}$$

Together with (107), this proves convergence (part 1), and, by proving boundedness of $\bar{\epsilon}_t \equiv \|\epsilon_{I_{\bar{f}_t}}\|_\infty$, completes the proof of the error bound (part 2).

It remains to combine convergence and error bound to obtain the joint convergence statement in part 3 of the proposition: Analogously to the proof of Proposition 1, one can show that if the number of quadrature and interpolation nodes are chosen as $n_Q = (n/T)^\theta$ and $n_I = (n/T)^{1-\theta}$, respectively, then

$$\exists k_t < \infty : \|\epsilon_{I_{\bar{f}_t}}\|_\infty \leq k_t \left(\frac{n}{T}\right)^{-s} \tag{125}$$

where $s = \min\{s_Q\theta, s_I(1-\theta)\}$ and $k_t \equiv 2\max\{k_t^*, k_t^{**}\}$ as defined above.

Since $\bar{T}, k_t < \infty$, a k can be chosen as $k \equiv \max_{t\in 1,\dots,\bar{T}} k_t$ and therefore

$$\exists k < \infty : \forall t \in 1,\dots,\bar{T} : \|\epsilon_{I_{\bar{f}_t}}\|_\infty \leq k \left(\frac{n}{T}\right)^{-s} \tag{126}$$

and in particular

$$\max_{t\in 1,\dots,\bar{T}} \|\epsilon_{I_{\bar{f}_t}}\|_\infty \leq k \left(\frac{n}{T}\right)^{-s} \tag{127}$$

Since I know from inequality (107) of the proof of part 2 (together with Equations 118 and 124) that the overall error can be bounded linearly, part 3 can be proved by noting that

$$\begin{aligned} |\epsilon_{L_f^T}| &\leq \sum_{t=1}^{T} \|\epsilon_{I_{\bar{f}_t}}\|_\infty && (128)\\ &\leq T \max_{t\in 1,\dots,\bar{T}} \|\epsilon_{I_{\bar{f}_t}}\|_\infty && (129)\\ &\leq Tk \left(\frac{n}{T}\right)^{-s} && (130) \end{aligned}$$

□

Remark 11 (Role of Assumptions and Restrictions in Proposition 2)**.** I list a couple of remarks on the role and limitations of the assumptions and restrictions necessary to prove Proposition

2:

Boundedness of Time Horizon The assumption that $T \leq \bar{T} < \infty$ has no practical relevance, as $\bar{T}$ can be chosen arbitrarily large, as long as it is finite. Rather, it is technically needed to bound the sum of errors independently of T (Equations 109 and 127), because the supremum of $\bar{\epsilon}_t$ w.r.t. t is not necessarily finite. This is because although f_t in Proposition 2 is restricted to be smaller or equal to 1, $\bar{f}_t$ as defined by Equation (90) not necessarily is, because there is no guarantee that the interpolant from the previous iteration, $\mathcal{I}_{\hat{I}_{\bar{f}_{t+1}}}$, actually respects the bound; consequently, since $\bar{f}_t$ is recursively multiplied against $\mathcal{I}_{\hat{I}_{\bar{f}_{t+1}}}$, it could—in theory—happen that $\bar{f}_t$ is unbounded for some t as T tends to infinity, rendering the convergence assumptions of the quadrature and interpolation methods in use invalid.

Boundedness of the Integrand Although the proof of Proposition 2 requires the integrand to map to $[0, 1]$, for practical purposes, the image space really only needs to be bounded (which is required by the definition of C^i anyway). Then, since integration is a linear operator, the integrand can be (linearly) transformed to comply with the restrictions, integrated recursively, and finally the integral is transformed back. Note that the same argument can be applied for kernels that integrate to more than one (c.f. Definition 1).

Smoothness Preservation of Interpolation Note that while the proof of Proposition 2 requires the interpolation method to be smoothness preserving as in Definition 11, it does not require the degree of smoothness preservation to be such that the *maximum* possible convergence rate will be attained; rather, different degrees of smoothness preservation are generally required for parts 1 and 2 (together), or 3 to hold. For example, it is well known that every continuous and bounded function over a compact interval is a uniform limit of piecewise linear continuous functions, and interpolation using piecewise linear continuous functions preserves continuity. At the same time, the trapezoidal rule for numerical integration converges for every Riemann integrable function, i.e. for all bounded and continuous functions over a compact domain, and the compactness of the domain for the interpolation assures uniformity of convergence of the parametric integration problem. Therefore, parts 1 and 2 of Proposition 2 still assure convergence and linear error growth.

Corollary 2 (Sub-Linear Error Bound)**.** *Given the approximation problem of Proposition 1 with the corresponding assumptions, suppose that the computational effort for each iteration implied by n_Q and n_I is fixed, and $T \leq \bar{T} < \infty$. Then there exists an error bound for $|\epsilon_{L_f^T}|$ that grows sub-linearly in T.*

Proof. To prove sub-linearity, note that there is a stricter bound to (106), namely

$$\sum_{t=1}^{T} |\epsilon_t| \leq \sum_{t=1}^{T} L_f^{t-1} \bar{\epsilon}_t \tag{131}$$

$$\leq \bar{\epsilon} \sum_{t=1}^{T} L_f^{t-1} \tag{132}$$

$$\equiv E_T \tag{133}$$

where $\bar{\epsilon} \equiv \max_{t \in 1,\dots,\bar{T}} \bar{\epsilon}_t$. Due to (101), I can write

$$E_T - E_{T-1} = \bar{\epsilon} \sum_{t=1}^{T} L_f^{t-1} - \bar{\epsilon} \sum_{t=1}^{T-1} L_f^{t-1} \tag{134}$$

$$= L_{T-1} \bar{\epsilon} \tag{135}$$

$$\leq L_{T-2} \bar{\epsilon} \tag{136}$$

$$= \bar{\epsilon} \sum_{t=1}^{T-1} L_f^{t-1} - \bar{\epsilon} \sum_{t=1}^{T-2} L_f^{t-1} \tag{137}$$

$$= E_{T-1} - E_{T-2} \tag{138}$$

which proves sub-linearity. □

Remark 12 (Sharpness of Error Bound)**.** Note that the error bounds of Proposition 2 and Corollary 2 are not sharp, since by taking absolute errors in (95), they ignore sign changes in $\epsilon_{I_{\bar{f}_t}}$ which potentially make the errors cancel or "average out". Rather, (sub)linear error growth is derived from the boundedness of the integrand by 1 in this paper. However, since the sign changes and their effect on the overall error depend on the function $\epsilon_{I_{\bar{f}_t}}$ and therefore on f itself, they are much harder to quantify ex ante.

I conclude the section with a couple of formal and numerical examples that demonstrate the different interpretations of the theoretical results.

Example 7 (Increasing Accuracy)**.** Suppose the level of accuracy for $\hat{L}_f^T$ shall be increased by a factor of i; one wants to know how many more integrand evaluations are necessary to obtain this level of accuracy. Therefore, the following equation needs to be solved for j:

$$T \left(\frac{n}{T}\right)^{-s} = iT \left(\frac{jn}{T}\right)^{-s} \tag{139}$$

where $s = \min\{s_Q \theta, s_I(1-\theta)\}$, yielding $j = i^{\frac{1}{s}}$. For example, to double the level of accuracy ($i = 2$) for a given time horizon T when approximating a one-dimensional integrand from C^4 using a cubic spline and Simpson integration ($s_Q = s_I = 4$, assuming that the third and forth derivative are approximated well enough to preserve smoothness) with equally many quadrature and interpolation nodes ($\theta = 0.5$), one needs to increase the total number of integrand evaluations by a factor of $2^{\frac{1}{2}} \approx 1.4$. In contrast, to double the average level of accuracy using Monte Carlo integration, 4 times more integrand evaluations are needed.

Example 8 (Convergence Rate Comparison)**.** This example compares the rates of convergence for different approximation methods for the recursive parametric integral from Definition 13 with a particular integrand f. More precisely, the approximation error is computed for different total numbers of integrand evaluations n, but a fixed time horizon T. To compare the recursive scheme (using any quadrature and interpolation method) to Monte Carlo integration, the following equality has to hold:

$$n = T n_Q n_I = T n_{MC} \tag{140}$$

where n_Q and n_I are the numbers of nodes for quadrature and interpolation in each iteration

of the recursion, respectively, and n_{MC} is the number of (T-dimensional) draws for the MC integration.

Consider the integrand

$$f : \mathbb{R}^2 \to [0,1],\ f(x_t, x_{t-1}) = \Phi((0.5x_{t-1} + x_t + 4)^2) \tag{141}$$

where Φ is the cumulative distribution function of the standard normal distribution, and set the kernel q equal to the density function of the standard normal distribution. (Note that this formulation embeds an $AR(1)$ process with normal innovations and persistence parameter $\rho = 0.5$; see below.)

The methods in this example are parametrized as follows: The first configuration of recursive parametric integration uses Gauss-Hermite quadrature and Chebyshev polynomials (see Examples 2 and 3), using $2n_Q^{GH} = n_I^{Cheb}$ since the convergence rate of Gaussian rules is twice as much as the best convergence rate of polynomial interpolation. The second configuration of the recursive method uses the compound Simpson rule and cubic spline interpolation (see Examples 1 and 5) with $n_Q^{Simp} = n_I^{CS}$. The time horizon is fixed at $T = 100$, and 100 MC integrations are performed to estimate the standard deviation of the MC estimate of L_f^T. The benchmark is computed using the recursive method with Gauss-Hermite quadrature and Chebyshev polynomials, with $n_Q^{GH} = 255$ and $n_I^{Cheb} = 511$.[13]

Figure 1 depicts the error of the two recursive parametric integration versions and MC integration as a function of integrand evaluations (normalized by T); since both axes are on a log scale, convergence rates can be read from the slope of the (log) error as a (linear) function of the (log) number of nodes or draws (and thus the number of integrand evaluations). As a visual support, two triangles depict slopes of $-1/2$ and -2, respectively. The example confirms the result from Proposition 2 (together with Remark 10), given the convergence rates of Gauss quadrature and polynomial interpolation are exponential at best, the rates of Simpson integration and cubic splines are both 4, and the standard deviation of Monte Carlo integration reduces at rate $1/2$ (see Remark 4). Note that although the cubic spline is not smoothness preserving in the sense of Definition 11, the convergence rates predicted by Proposition 2 can still be observed; therefore, none of these requirements are necessary, but, by proof, they are sufficient.

Figure 2 plots the runtime in seconds needed to achieve a particular degree of numerical accuracy (again both in log terms). Since MC integration is naturally faster for a comparable total number of integrand evaluations (because no interpolant creation and evaluation is needed), but has a far slower convergence, only by putting the runtime of the integration into relation with the numerical accuracy achieved within a particular amount of time, one obtains a realistic perception of the true computational efficiency of the methods. In this particular example, it is obvious that while MC integration is far more efficient to obtain a rough estimate of the integral, the recursive method is way more efficient to obtain accurate approximations of the integral,

[13] In order to compute the numerical errors $|L_f^T - \hat{L}_f^T|$ for each configuration, the true but generally unknown solution L_f^T is approximated by the recursive method, but using very large numbers of quadrature and integration nodes, respectively. Due to the convergence result of Proposition 1, this is a valid approach; the particular numbers for n_Q and n_I are obtained from exponentially increasing them until the relative error is below a certain threshold, which is a widely used practice in numerical analysis.

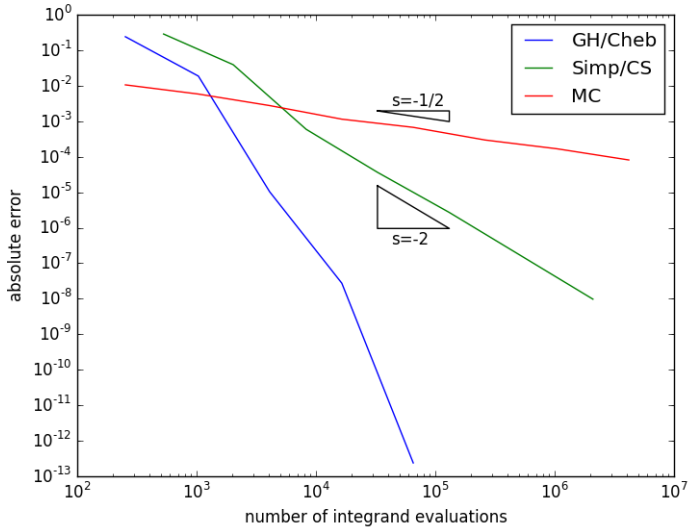


Figure 1: Numerical approximation error of recursive parametric integration using Gauss-Hermite integration with Chebyshev interpolation, and Simpson integration and cubic spline interpolation, respectively, compared to Monte Carlo integration, as a function of the total number of integrand evaluations.

which can be highly beneficial when put into an optimization context (see Section 3.4 below).[14]

Finally, Figure 3 depicts the compound interpolant over the iterations of the recursion. The figure gives a visual rationale why interpolation by (piecewise) polynomials works so well in this context.

Example 9 (Increasing T with Fixed Accuracy)**.** Suppose the time horizon increases from T to iT, one wants to know how many more integrand evaluations are needed to approximate L_f^{iT} to the same level of accuracy. Therefore, one needs to solve the following equation for j:

$$T\left(\frac{n}{T}\right)^{-s} = iT\left(\frac{jn}{iT}\right)^{-s} \tag{142}$$

where $s = \min\{s_Q\theta, s_I(1-\theta)\}$, yielding $j = i^{\frac{s+1}{s}}$. For example, to double T ($i = 2$) when approximating a one-dimensional integrand from C^4 using the same configuration as in Example 7, $2^{\frac{3}{2}} \approx 2.8$ times more integrand evaluations are necessary to obtain a comparable level of accuracy. This is in contrast to Monte Carlo integration, which yields—on average—a level of error that is independent of T, and thus only the additional evaluations of the integrand for $t > T$ need to be accounted for. Therefore, MC uses only 2 times as many integrand evaluations. Note that as s grows larger, the recursive integration scheme becomes more and more independent of T, similarly to Monte Carlo, since $\lim_{s\to\infty} j^{\frac{s+1}{s}} = j$, which can be observed when using methods with exponential convergence.

[14] All examples in this section are written in Python (partially using Numpy and Scipy), without any parallelization. All computations are carried out on a 2012 laptop with one four core Intel "Core i7 Ivy Bridge" processor running at 2.6 GHz and 16GB RAM.

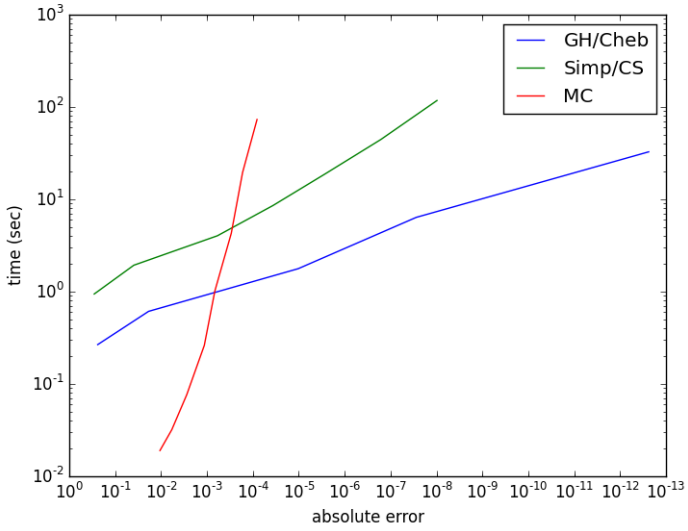


Figure 2: Runtimes (in seconds) of recursive parametric integration using Gauss-Hermite integration with Chebyshev interpolation, and Simpson integration and cubic spline interpolation, respectively, compared to Monte Carlo integration, as a function of numerical accuracy.

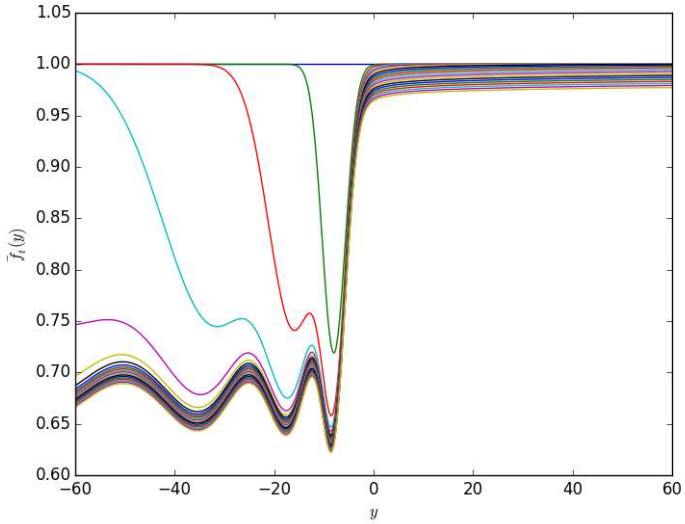


Figure 3: Interpolant of recursive likelihood function integration, $\bar{f}_t$, during the recursion, $t = 20, \ldots, 1$ (from top to bottom; $\bar{f}_{20}$ corresponds to constant 1).

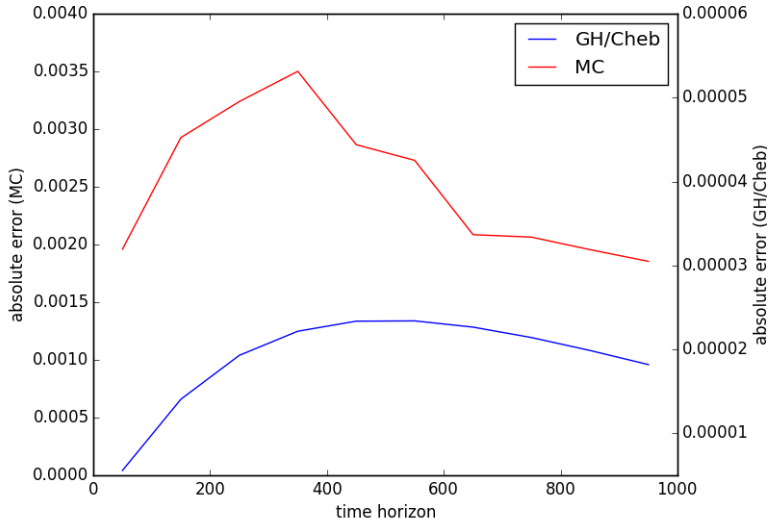


Figure 4: Numerical approximation error of recursive parametric integration using Gauss-Hermite integration with Chebyshev interpolation (right axis) and Monte Carlo integration (left axis) for different time horizons T.

Example 10 (Error Growth in T)**.** This example investigates the accumulation of the error in L_f^T as a function of T for the same integrand as in Example 8, given the computational resources for each iteration of the recursive integral and the number of (T-dimensional) draws are fixed to n_Q, n_I, and n_{MC}, respectively. By Corollary 2, the approximation error of the recursive method is expected to grow sub-linearly in T.

For the numerical approximation, I use Gauss-Hermite quadrature with 45 nodes, and Chebyshev polynomials of degree 90, and Monte Carlo integration with the corresponding number of draws. The benchmark is computed using the same configuration as in Example 8, and 100 MC simulations are run to obtain an estimate of the standard deviation.

Figure 4 plots the absolute approximation error $|L_f^T - \hat{L}_f^T|$ for different values of T; note that in this example, the error is bounded independently of T, which further indicates that not even the sub-linear error bound from Corollary 2 is sharp (recall that Corollary 2 only bounds sub-linearly), presumably because it ignores the effect of errors "averaging out" in the integration through its bounding by absolute values, as pointed out in Remark 12. This is particularly plausible, as a similar effect might also be inherent to Monte Carlo integration, and the results are qualitatively comparable. Note that the error of Monte Carlo integration is measured on the left axis, while the error for the recursive method is measured on the right axis.

Figure 5 plots the runtime for both methods as a function of the time horizon, and confirms the linear complexity in T for both methods.

I conclude the discussion of the convergence behavior of recursive parametric integration by applying it to the Rust (1987) model with serially correlated unobserved errors as defined in Section 2.1. (See Section 3.4 below on when and how to apply recursive parametric integration to a likelihood function.)

Example 11 (Likelihood Function of the Rust (1987) Model with Serial Correlation)**.** This

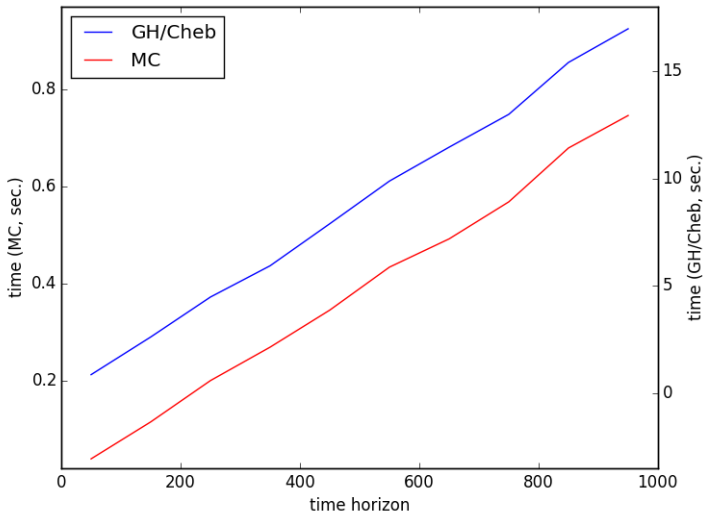


Figure 5: Runtimes (in seconds) of recursive parametric integration using Gauss-Hermite integration with Chebyshev interpolation (right axis) and Monte Carlo integration (left axis) for different time horizons T.

example analyses the convergence properties of the likelihood function of the Rust (1987) model with serially correlated unobserved errors as defined in Section 2.1 (in particular Equation 11).[15] The main objective of this example is to demonstrate the convergence behavior of the recursive parametric integration method when the conditions on smoothness and its preservation fail to hold.

My implementation of the Rust (1987) model with serially correlated unobserved errors—which is outlined in detail in Appendix A.1—has two potential sources of non-smoothness: First, to approximate the expected value function (8), I use a piecewise linear interpolation scheme with grid adaption; while this is a generic and numerically robust approach, it generally creates interpolants that are continuous only, but not smooth. Since the EV function enters the choice probabilities, which are finally integrated w.r.t. the serially correlated errors, the actual integrand might not be as smooth as required by Proposition 2. Second, the actual methods employed for the quadrature are Gauss–Hermite quadrature, paired with splines (linear (i.e. PWL), cubic, and Akima); since splines are neither fully smoothness preserving in the sense of Definition 11 (see Examples 4 and 5), nor do they deliver sufficiently smooth integrands for the Gauss quadrature to exhibit exponential convergence even at high orders in theory (see Example 2), I will first analyze the convergence of the quadrature method in isolation, to relate it to the results above.

Figure 6 plots the absolute approximation error of the integrated likelihood function, as a function of different number of quadrature nodes for each iteration, n_Q, but for a fixed number of interpolation nodes, n_I. This function is plotted for three different levels of grid adaption errors (distinguished by color), as well as two numbers of interpolation nodes, n_I (distinguished by line

[15] The actual parametrization is chosen to match the MLE with normal innovations (not normalized) for bus groups $\{1, 2, 3, 4\}$ (see Table 3).

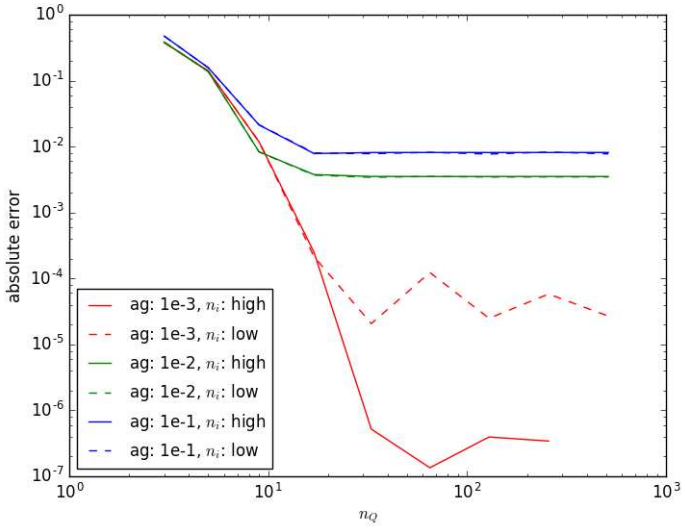


Figure 6: Numerical approximation error of recursive parametric integration of the likelihood function of the Rust (1987) model with serially correlated errors (see Section 2.1), using Gauss-Hermite integration with Akima spline interpolation, as a function of the number of quadrature nodes per iteration, n_Q. The function is plotted for three different levels of accuracy of the EV function approximation (labeled "ag" in the legend, smaller is better; distinguished by color), and two different numbers of interpolation nodes per iteration, n_I ("low": $n_I = 1,000$, "high": $n_I = 8,000$; distinguished by line type).

type). Two facts are noteworthy: First, the minimum attainable error is mostly determined by the degree of refinement of the EV function; this is not surprising, as the EV function itself is smooth, and the piecewise linear approximation together with the grid adaption results in an arbitrarily close approximation quite fast. On the other hand, if the approximation of the EV function is itself is not accurate, the accuracy of the approximation of the likelihood function cannot be increased beyond some level, because, loosely speaking, it is the likelihood of a different EV function—namely the poor approximation of the true EV function—that is approximated to very high precision. Therefore, computing an accurate approximation of the input is obviously a necessary condition for accurate output. Second, given the approximation of the EV function is sufficiently precise, the minimum attainable error is bound by the number of interpolation nodes; this is expected, and fully in line with the theoretical results from Proposition 2, because, intuitively, if not both the number of quadrature and the number of interpolation nodes are increased simultaneously, increasing only one will finally result in approximating the wrong target rather than increasing accuracy (similar to the argument above). That being said, I find that although the requirements on smoothness and its preservation are not fulfilled in this example, the convergence of the quadrature method is still exponential (before flattening out for the reasons mentioned above); therefore, it is clear that if the interpolation method in use is converging at a polynomial rate only, "almost all" nodes should be spend for interpolation rather than quadrature (see Remark 10).

Figure 7 depicts the absolute approximation error of the integrated likelihood function, as a function of different number of interpolation nodes for each iteration, n_I. As argued in

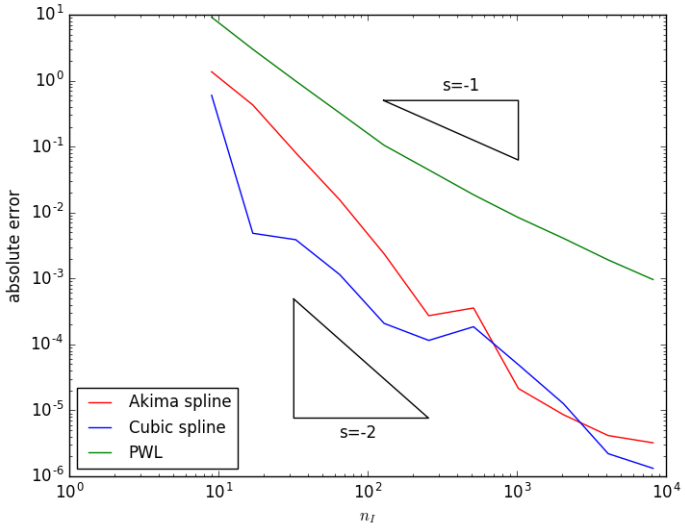


Figure 7: Numerical approximation error of recursive parametric integration of the likelihood function of the Rust (1987) model with serially correlated errors (see Section 2.1), using Gauss-Hermite integration with different kind of spline interpolation (distinguished by color), as a function of the number of interpolation nodes per iteration, n_I.

the previous paragraph, the vastly faster convergence of the quadrature method (exponential) compared to splines (polynomial at rates between 2 and 4, if smoothness conditions apply) leads me to fix the number n_Q and only increase n_I, so that $n_I \gg n_Q$, as well as to choose a high level of accuracy for the EV function; the fastest convergence I can expect according to Proposition 2 together with Remark 10 is therefore s_I. However, looking at the figure, it appears that the maximum convergence rate of the respective interpolation methods cannot be attained; in particular, the rates seem to "flatten out" a bit for higher n_Q. While the general observation that the maximum convergence rate of interpolation cannot be attained likely stems for the fact that the interpolation input is not smooth enough do to a lack of smoothness preservation, the flattening might be related also to the limited precision of the EV approximation, as discussed above.

3.4 Recursive Likelihood Function Integration

Definition 14 (Model)**.** Suppose the model under consideration predicts observations according to the joint probability density function[16]

$$P_{xi\varepsilon}(\{i_t, x_t, \varepsilon_t\}_{t=1}^{T} | \{i_0, x_0, \varepsilon_0\}; \theta) \tag{143}$$

where i_t is the dependent variable observed at time t ("outcome"), x_t and ε_t are the observable and the unobservable parts of the independent variables at time t, respectively, and θ is a vector of parameters of the model.

Note that i_t, x_t, and ε_t in Definition 14 are generally vector-valued.

[16]Without explicitly mentioning, I assume that all density functions referred to throughout the paper exist.

Assumption 1 (Markov Property)**.** *The model in Definition 14 is Markov:*[17]

$$P_{xi\varepsilon}(\{i_t, x_t, \varepsilon_t\}_{t=1}^T | \{i_0, x_0, \varepsilon_0\}; \theta) = \prod_{t=1}^T p_{xi\varepsilon}(\{i_t, x_t, \varepsilon_t\} | \{i_{t-1}, x_{t-1}, \varepsilon_{t-1}\}; \theta) \tag{144}$$

Rewriting the transition probability density function using conditional density functions yields

$$\begin{aligned} &p_{xi\varepsilon}(\{i_t, x_t, \varepsilon_t\} | \{i_{t-1}, x_{t-1}, \varepsilon_{t-1}\}; \theta) \\ &\quad = p_{xi|\varepsilon}(i_t, x_t | i_{t-1}, x_{t-1}, \varepsilon_{t-1}, \varepsilon_t; \theta) p_\varepsilon(\varepsilon_t | i_{t-1}, x_{t-1}, \varepsilon_{t-1}; \theta). \end{aligned} \tag{145}$$

Assumption 2 (Smoothness of Conditional Density Functions)**.** *The conditional density functions $p_{xi|\varepsilon}(i_t, x_t | i_{t-1}, x_{t-1}, \varepsilon_{t-1}, \varepsilon_t; \theta)$ and $p_\varepsilon(\varepsilon_t | i_{t-1}, x_{t-1}, \varepsilon_{t-1}; \theta)$ are in C^i w.r.t. to ε_t and ε_{t-1}.*

Assumption 3 (Boundedness of Conditional Density Function for Observed Variables)**.** *The conditional density function of the observed variables, $p_{xi|\varepsilon}(i_t, x_t | i_{t-1}, x_{t-1}, \varepsilon_{t-1}, \varepsilon_t; \theta)$, is bounded by 1.*

Since the process ε_t is unobserved, the likelihood function of model (143) cannot be computed directly. However, the *marginal* likelihood function computes as the marginalization with respect to $\{\varepsilon_t\}_{t=0}^T$:[18]

Definition 15 (Marginal Likelihood Function)**.** The *marginal* likelihood function of model (143) reads as

$$L_T(\theta) \equiv P_{xi}(\{i_t, x_t\}_{t=0}^T | \{i_0, x_0\}; \theta) \tag{146}$$

$$= \int_{\underline{\varepsilon}_0}^{\bar{\varepsilon}_0} \cdots \int_{\underline{\varepsilon}_T}^{\bar{\varepsilon}_T} p_\varepsilon(\varepsilon_0; \theta) \prod_{t=1}^T p_{xi|\varepsilon}(i_t, x_t | i_{t-1}, x_{t-1}, \varepsilon_{t-1}, \varepsilon_t; \theta) p_\varepsilon(\varepsilon_t | i_{t-1}, x_{t-1}, \varepsilon_{t-1}; \theta) d\varepsilon_0 \dots d\varepsilon_T \tag{147}$$

Note that since ε_t in (147) can be vector valued, the integrals over each ε_t form potentially multi-dimensional integrals themselves.

In order to allow for random variables with infinite support (without truncation), the integral in the marginal likelihood function (146) has to be transformed into a kernel integral in the sense of Definition 1. Therefore, I require the following assumption:

Assumption 4 (Change of Variable)**.** *There exists an invertible change of variable*

$$\varepsilon_t = \varphi(\tilde{\varepsilon}_t, \varepsilon_{t-1}, i_{t-1}, x_{t-1}; \theta) \tag{148}$$

such that

$$p_\varepsilon(\varphi(\tilde{\varepsilon}_t, \varepsilon_{t-1}, i_{t-1}, x_{t-1}; \theta) | i_{t-1}, x_{t-1}, \varepsilon_{t-1}; \theta) = q_t(\tilde{\varepsilon}_t; \theta) \tag{149}$$

[17] In this paper, I only consider Markov models of order 1; while the RLI method extends to higher orders, the convergence rates of Sections 3.2 and 3.3 have to be generalized to apply to higher order Markov models.

[18] I use the term "marginal likelihood function" in this context in the frequentist's sense, in that the unobserved random variablesx ε_t can be thought of as *nuisance parameters* with a distribution attached to them, which allows to integrate them out (instead of being optimized over).

which is in C^{i+1} *w.r.t.* $\tilde{\varepsilon}_t$ *and* ε_{t-1}*, and where*

$$\int_{\varphi^{-1}(\underline{\varepsilon}_t,\varepsilon_{t-1},i_{t-1},x_{t-1};\theta)}^{\varphi^{-1}(\bar{\varepsilon}_t,\varepsilon_{t-1},i_{t-1},x_{t-1};\theta)} q_t(\tilde{\varepsilon}_t;\theta)d\tilde{\varepsilon}_t \leq 1 \tag{150}$$

$$\varphi'(\tilde{\varepsilon}_t,\varepsilon_{t-1},i_{t-1},x_{t-1};\theta) \equiv \frac{\partial}{\partial\tilde{\varepsilon}_t}\varphi(\tilde{\varepsilon}_t,\varepsilon_{t-1},i_{t-1},x_{t-1};\theta) \leq 1 \tag{151}$$

Example 12 ($AR(1)$ Process)**.** The change of variable φ in Assumption 4 often coincides with the "functional form" of the process ε_t. A simple but practically very important example for a particular process and the corresponding change of variable is the $AR(1)$ process

$$\varepsilon_t = \rho\varepsilon_{t-1} + \tilde{\varepsilon}_t \tag{152}$$

where $\tilde{\varepsilon}_t$ is white noise, distributed identically and independently according to the density function $q(\cdot)$, and $\varepsilon_t = 0$ for $t \leq 0$. The corresponding change of variable that fulfills (149) is

$$\varphi(\tilde{\varepsilon}_t,\varepsilon_{t-1},i_{t-1},x_{t-1};\theta) = \varphi(\tilde{\varepsilon}_t,\varepsilon_{t-1};\theta) = \rho\varepsilon_{t-1} + \tilde{\varepsilon}_t \tag{153}$$

$$\varphi'(\tilde{\varepsilon}_t,\varepsilon_{t-1},i_{t-1},x_{t-1};\theta) = 1 \tag{154}$$

$$\varphi^{-1}(\varepsilon_t,\varepsilon_{t-1},i_{t-1},x_{t-1};\theta) = \varepsilon_t - \rho\varepsilon_{t-1} \tag{155}$$

Proposition 3 (Recursive Likelihood Function Integration)**.** *Given a marginal likelihood function* $L_T(\theta)$ *as in Definition 15, consider its approximation by recursive application of the parametric integral approximation using quadrature and interpolation as in Definition 12, where the quadrature and the interpolation methods converge uniformly at rates* s_Q *and* s_I *in the sense of Definitions 8 and 10 for sufficiently smooth integrands, respectively, and where the interpolation method is moreover smoothness preserving as by Definition 11. Under Assumptions 1, 2, 3, and 4, the results of Proposition 2 as well as Corollary 2 apply.*

Proof. Assumptions 2 and 4 ensure that integration by substitution is valid, and thus the integrand in (147) can be written as

$$\begin{aligned}
&\int_{\underline{\varepsilon}_t}^{\bar{\varepsilon}_t} p_{xi|\varepsilon}(i_t,x_t|i_{t-1},x_{t-1},\varepsilon_{t-1},\varepsilon_t;\theta)p_\varepsilon(\varepsilon_t|i_{t-1},x_{t-1},\varepsilon_{t-1};\theta)d\varepsilon_t \\
&= \int_{\varphi^{-1}(\underline{\varepsilon}_t,\varepsilon_{t-1},i_{t-1},x_{t-1};\theta)}^{\varphi^{-1}(\bar{\varepsilon}_t,\varepsilon_{t-1},i_{t-1},x_{t-1};\theta)} p_{xi|\varepsilon}(i_t,x_t|i_{t-1},x_{t-1},\varepsilon_{t-1},\varphi(\tilde{\varepsilon}_t,\varepsilon_{t-1},i_{t-1},x_{t-1};\theta);\theta) \\
&\quad \cdot \varphi'(\tilde{\varepsilon}_t,\varepsilon_{t-1},i_{t-1},x_{t-1};\theta)q_t(\tilde{\varepsilon}_t;\theta)d\tilde{\varepsilon}_t
\end{aligned} \tag{156}$$

This change of variable can be applied to the marginal likelihood function (146), omitting

the integration limits for better readability:

$$L_T(\theta) = \int \cdots \int q_0(\tilde{\varepsilon}_0;\theta) \tag{157}$$

$$\cdot \prod_{t=1}^{T} \underbrace{p_{xi|\varepsilon}(i_t, x_t | i_{t-1}, x_{t-1}, \varepsilon_{t-1}, \varphi(\tilde{\varepsilon}_t, \varepsilon_{t-1}, i_{t-1}, x_{t-1};\theta);\theta)\varphi'(\tilde{\varepsilon}_t, \varepsilon_{t-1}, i_{t-1}, x_{t-1};\theta)}_{\equiv f_t(\tilde{\varepsilon}_t, \varepsilon_{t-1})}$$

$$\cdot q_t(\tilde{\varepsilon}_t;\theta) d\tilde{\varepsilon}_0 d\tilde{\varepsilon}_1 \dots d\tilde{\varepsilon}_T \tag{158}$$

$$\equiv \int \cdots \int q_0(\tilde{\varepsilon}_0) \prod_{t=1}^{T} f_t(\tilde{\varepsilon}_t, \varepsilon_{t-1}) q_t(\tilde{\varepsilon}_t) d\tilde{\varepsilon}_0 d\tilde{\varepsilon}_1 \dots d\tilde{\varepsilon}_T \tag{159}$$

$$= L_f^T \tag{160}$$

where $q_t(\cdot) \equiv q_t(\cdot;\theta)$. Note that the substitution by f_t is possible since i and x are observed for all t.

Most importantly, note that due to the fact that $p_{xi|\varepsilon} \leq 1$ by Assumption 3, and due to Assumption 4, the integrand f_t is restricted to $[0,1]$. Therefore, and since $f_t \in C^i$ due to $p_{xi|\varepsilon}, p_\varepsilon \in C^i$ w.r.t. ε_t and ε_{t-1}, and $\varphi, \varphi', \in C^i$ w.r.t. $\tilde{\varepsilon}_t$ and ε_{t-1}, given all other assumptions of this proposition hold, Proposition 2 as well as Corollary 2 apply. □

Remark 13 (Role of Assumptions in Proposition 3)**.** I list a couple of remarks on the role and limitations of the assumptions necessary for Proposition 3 to apply:

Continuity of Density Functions Assumption 2 is not always fulfilled by default and needs some care in model design; for example, in discrete choice models it can happen that that the *conditional* choice probability is binary and thus degenerate: $p_{i|x\varepsilon}(i_t|x_t, \varepsilon_t, \theta) \in \{0,1\}$ which is not even continuous; however, there exist ways to avoid this kind of problem, such as introducing smooth (uncorrelated) errors, etc.

Unit Bound on Density Function While the unit bound restriction of Assumption 3 is technically needed to prove Proposition 3 below, it is rarely restricting in the practice of maximum likelihood estimation, since even if it fails to hold, any monotone transformation of the likelihood function—such as rescaling $p_{xi|\varepsilon}$—will preserve the location of its maximum. The same holds true for the unit bound on the kernel integral and the corresponding change of variable in Equations (150) and (151) below. Alternatively, the rescaling can be done within the numerical integration, as noted in Remark 11.

Change of Variable Assumption 4 is w.l.o.g., because the use of the kernel integral as in Definition 1 is itself w.l.o.g.; numerically, even non-trivial kernels can always be made part of the integrand f, speaking in terms of Definition 1. However, if expectations over random variables with infinite support are integrated, either the kernel integral has to be used, or the domain of integration has to be truncated. Even if, however, the change of variables is necessary, the unit boundedness in Equations (150) and (151) is also w.l.o.g as argued in Remark 11.

4 Estimation Results for the Bus Engine Replacement Model

In this section, I estimate the bus engine replacement model of Rust (1987) featuring a serially correlated, unobserved random utility component, as specified in Section 2.1. First, I present the estimation results for the original dataset in Section 4.1; second, I carry out an extensive Monte Carlo study with simulated datasets in Section 4.2, to assess the question to which extend the algorithm is able to reproduce the parameters of a distribution with known parameters, and what estimator variance can be expected from various dataset sizes.

4.1 Original Dataset

The original dataset of Rust (1987) consists of monthly odometer readings and engine replacement decisions for a fleet of 162 buses, subdivided into 8 groups depending on their manufacturer and model. Since buses are heterogeneous across groups, it is common to create different subsamples to estimate the parameters of model (1); I follow the literature by estimating three subsamples separately, consisting of groups $\{1, 2, 3\}$, $\{1, 2, 3, 4\}$, and $\{4\}$. Table 1 shows the size of the panel for each group under consideration.

Bus group	Number of buses (M)	Observation horizon (months)	Total number of observations	Number of replacements
1	15	25	360	0
2	4	49	192	0
3	48	70	3,312	27
4	37	117	4,292	33
Total	104		8,156	60

Table 1: Number of buses, observation time horizon in months, total number of observations, and number of observed engine replacements for each bus group.

As in Rust (1987), I discretize mileage in "bins" of 5,000 miles each.[19] The highest possible mileage state is 90 (which corresponds to 450,000 miles),[20] formally $x \in X = \{1, \ldots, 90\}$. I assume the mileage transition to follow a Markov process (conditional on the replacement decision), for which I estimate the parameters independently. I parametrize the discount factor by $\beta = 0.9999$ as in the original paper.

Before presenting the results of the estimation with serial correlation in the errors, I verify the estimation procedure presented in Section 2.2: Table 2 presents a partial reproduction of Table IX of Rust (1987), without serial correlation, but still numerically integrating both the expected value and the likelihood function. I conclude that, for the case without serial correlation, I am well able to replicate the original estimates.

Table 3 presents the estimation results using the original dataset of Rust (1987), again for

[19] By discretizing into bins of 5,000 miles I mean that the original mileage $\tilde{x}$ transforms into a mileage state $x = \lceil \tilde{x}/5,000 \rceil$, with the ceiling function $\lceil \tilde{y} \rceil = \min\{y \in \mathbb{N} : y \geq \tilde{y}\}$.

[20] If a bus ever reaches the maximum mileage state, I assume it to stay there until engine replacement. Although no bus in any of the subsamples ever reaches the maximum mileage state, it still has relevance for the solution of the dynamic problem of the agent, who takes this possibility into account when solving his infinite horizon dynamic optimization problem.

	Bus groups 1–3		Bus groups 1–4		Bus group 4	
	Rust (1987)	Estimated	Rust (1987)	Estimated	Rust (1987)	Estimated
RC	11.7270	11.7266	9.7558	9.7560	10.0750	10.0749
	(2.602)	(1.928)	(1.227)	(0.898)	(1.582)	(1.351)
θ_1	4.8259	4.8257	2.6275	2.6276	2.2930	2.2929
	(1.792)	(1.366)	(0.618)	(0.469)	(0.639)	(0.554)
ρ	–	–	–	–	–	–
L	-2,708.366	-2,708.366	-6,055.250	-6,055.250	-3,304.155	-3,304.156

Table 2: Replication of Table IX of Rust (1987) for all subsamples reported therein; L is the value of the log-likelihood function at the solution; $\beta = .9999$.

both extreme value type I and normally distributed innovations $\tilde{\varepsilon}$, and for each distribution family with and without normalization of the innovation distribution. In particular, an $AR(1)$ process with "standard" $EV1$ innovations with density $EV1(-\gamma, 1)$[21] will have mean 0 and variance $\pi^2/6(1-\rho^2)^{-1}$, while with normalized innovations, i.e. if innovations are distributed according to the density $EV1(-\gamma\sqrt{1-\rho^2}, \sqrt{1-\rho^2})$, the corresponding mean and variance will be 0 and $\pi^2/6$, respectively; for normal innovations, the $AR(1)$ process without normalization, i.e. with $N(0,1)$ innovations, will have zero mean, and variance $(1-\rho^2)^{-1}$, whereas the normalized version with $N(0, 1-\rho^2)$ innovations has mean zero and variance one. For the estimation of the standard errors from the original data set, I use the inverse of the negative Hessian of the likelihood function at its maximum, $(-H(\hat{\theta} \mid \{x_t, i_t\}_{t=0}^T))^{-1}$.[22]

For the $EV1$ case, I observe that while the parameter estimates in the presence of serial correlation are substantially different from the estimates without serial correlation, the ratio of engine replacement cost to the regular maintenance cost parameter is relatively stable; thus, the trade-off for the decision maker has not changed much quantitatively. This result holds true for both innovation specifications, i.e. with and without normalization, although the values of the parameters in the normalized version are somewhere in between the values without serial correlation and the values with serial correlation but no normalization. Moreover, the relative costs and the corresponding likelihood function values are almost identical for the two specifications with serial correlation. Performing a likelihood ratio test to compute the statistical significance of the quantitative changes induced by the introduction of serial correlation, I find that only on the largest subsample of the dataset (bus groups 1–4) can I reject the hypothesis of no serial correlation at a reasonable significance level.

The case of normally distributed $\tilde{\varepsilon}$ yields similar results, with two notable differences: First, not only do the cost parameter values change substantially, but also are the ratios and thus the trade-off for the decision maker more distinct. However, at the same time, carrying out a likelihood ratio test, I cannot reject the hypothesis of no serial correlation at a reasonable

[21] The extreme value type I distribution, which is also sometimes also referred to as the Gumbel distribution, has as location and a scale parameter, μ and $\beta > 0$, and I denote its density by $EV1(\mu, \beta)$ if the parameters matter. The mean of a random variable with $EV1$ distribution is $\mu + \gamma\beta$, and its variance is $\beta^2\pi^2/6$, where $\gamma \approx 0.5772$ is the Euler-Mascheroni constant.

[22] Since all derivatives are approximated using finite differences, the numerical accuracy—in particular of second derivatives—is limited.

significance level for any of the subsamples in the normal case. Second, the density normalization has very little influence on the parameter estimates in the normal case, in contrast to the $EV1$ case where the difference is substantial.

I interpret the change of the ratio of the cost parameters in this particular model as follows (as an example, I assume the ratio in the restricted model to be larger than in the unrestricted one): If I ignore serial correlation, the relative costs of regular maintenance are underestimated. Consequently, using the true relative costs in a model without serial correlation, I would predict more (or, equivalently, earlier) engine replacement than I find in the data. Thus, allowing for serial correlation explains why I do not observe more frequent engine replacement, given the high (true) relative costs of regular maintenance. Conversely, in a model with serial correlation, but based on the biased relative costs estimates, I would predict the buses to run for too long without engine replacement.

Assessing the question of the statistical significance of the estimates from the original data set is difficult though. First, as I will demonstrate in the Monte Carlo study below, my experiments with artificial data sets indicate that the results are rarely significant for small samples, even if the true model features serial correlation as defined by (7). Consequently, given the number of buses in the original data set, p-values as for groups 1–4 with extreme value distributed $\tilde{\varepsilon}(i)$ is not what I can generally expect. Second, I still cannot conclude that the serial correlation I found in the data is really coming from an unobserved source, as different bus groups are pooled together for two of the three subsamples, thus creating a heterogeneous sample that is treated as homogeneous by the model. Consequently, as long as I do not find the serial correlation *within* one single bus group to be significant, these estimations have to be taken with a grain of salt.

4.2 Monte Carlo Study

In this section, I carry out an extensive Monte Carlo study, where I simulate the model from Section 2.1 to create many data sets of different sizes (number of buses),[23] for both "standard" densities, extreme value type I, $EV1(-\gamma, 1)$, and standard normal, $N(0, 1)$, and estimate the parameters from these data sets using NFXP together with the RLI algorithm. The objective is to investigate the ability of the method to recover the parameters from the data, for which I know the true values in the case of simulated data.[24] Therefore, for each data set size $M \in \{100\,,\, 1000\,,\, 10000\}$, and for both densities, I create 200 datasets; on each data set, I run an estimation with and without allowing for serial correlation (i.e. setting $\rho = 0$). Table 4 presents the results of this Monte Carlo study by reporting means and standard deviations of the respective estimates. I also report mean and standard deviation of the likelihood ratio test with the null hypothesis of absence of serial correlation, carried out on the individual data set level. Figures 8 and 9 finally plot a kernel smoothing estimation of the distribution of the estimates, together with the true parameter values, and the density of the normal distribution with mean and standard deviation as reported in Table 4.

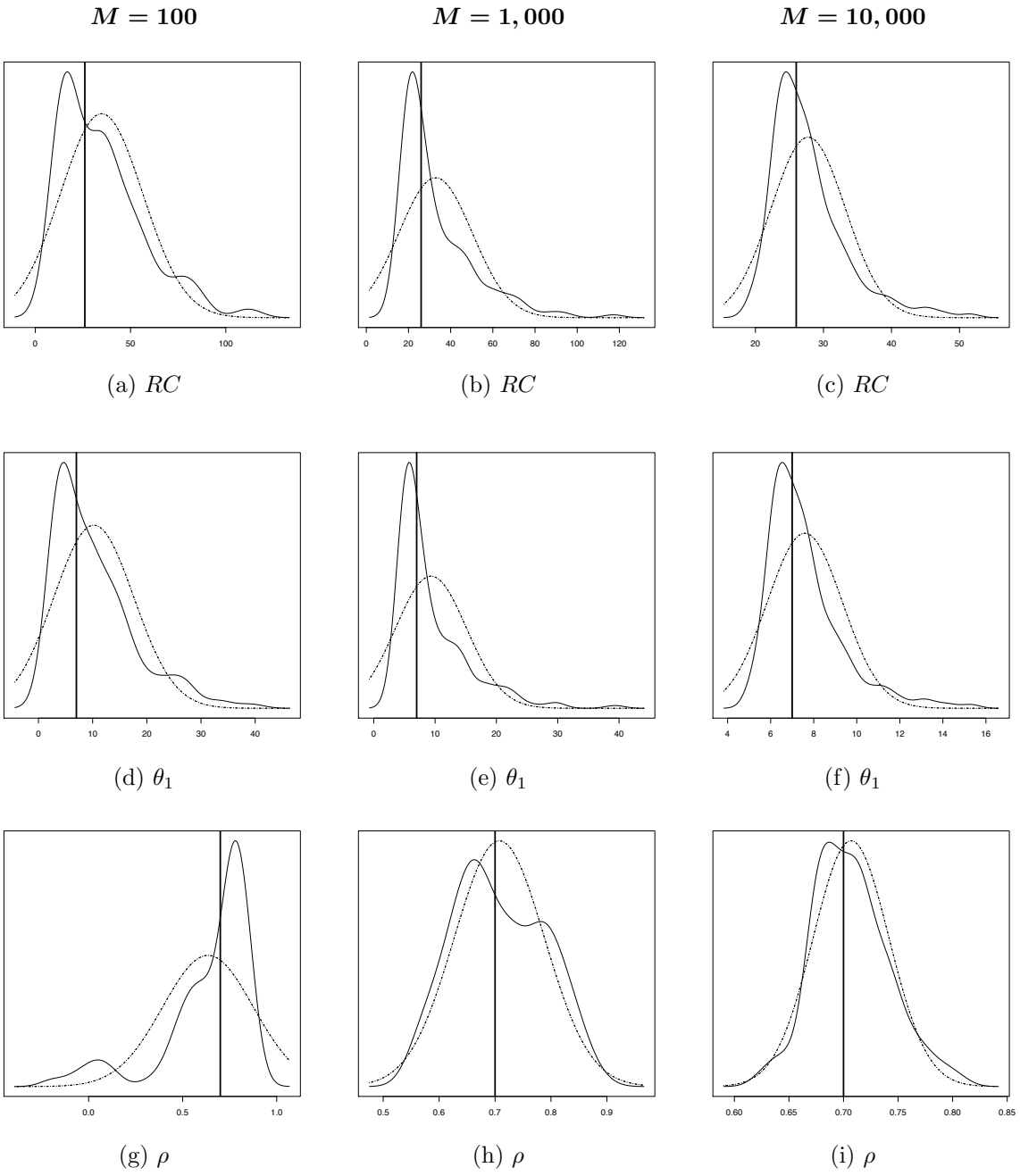
From this Monte Carlo study I draw the following conclusions: First, while the method

[23] Note that I refer to data set size as the number of buses in a dataset, or, equivalently, as the number of replacement observations, as I simulate each bus until replacement.

[24] The values for the parameters are chosen such that they resemble the estimates for the largest subset of the original dataset for the respective distribution, as reported below.

	$\tilde{\varepsilon} \sim EV1(\mu, \beta)$								
	Bus groups 1–3			Bus groups 1–4			Bus group 4		
		standard	normalized		standard	normalized		standard	normalized
RC	11.7266	25.0624	18.1397	9.7560	27.0368	18.19269	10.0749	22.0634	15.7403
	(1.928)	(10.127)	(10.859)	(0.898)	(16.939)	(6.724)	(1.351)	(13.873)	(9.452)
θ_1	4.8257	9.8531	7.1312	2.6276	7.4151	4.9894	2.2929	4.8132	3.4330
	(1.366)	(4.564)	(5.591)	(0.469)	(5.551)	(2.424)	(0.554)	(3.738)	(2.610)
RC/θ_1	2.4300	2.5396	2.5437	3.7128	3.6462	3.6463	4.3935	4.5840	4.5850
ρ	–	0.6899	0.6898	–	0.7396	0.7396	–	0.7001	0.7000
		(0.098)	(0.168)		(0.112)	(0.091)		(0.134)	(0.189)
L	-2,708.366	-2,707.777	-2,707.777	-6,055.250	-6,053.340	-6,053.340	-3,304.156	-3,303.912	-3,303.912
p (LR)		0.2903	0.2903		0.0506	0.0506		0.4848	0.4848
	$\tilde{\varepsilon} \sim N(0, \sigma)$								
RC	7.0372	13.8320	13.3909	6.0018	18.8660	17.5170	6.0747	10.8680	10.8111
	(1.029)	(2.736)	(0.552)	(0.481)	(2.671)	(2.777)	(0.758)	(0.948)	(4.056)
θ_1	2.5406	5.3814	5.4492	1.3990	5.2595	5.0840	1.1829	2.2881	2.4086
	(0.732)	(1.316)	(0.192)	(0.263)	(0.940)	(0.816)	(0.327)	(0.319)	(1.099)
RC/θ_1	2.7700	2.5717	2.4574	4.2900	3.5870	3.4455	5.1354	4.7497	4.4886
ρ	–	0.5203	0.5117	–	0.6680	0.6510	–	0.4887	0.4920
		(0.086)	(0.012)		(0.042)	(0.049)		(0.032)	(0.164)
L	-2,707.901	-2,707.832	-2,707.817	-6,054.082	-6,053.683	-6,053.649	-3,303.919	-3,303.899	-3,303.889
p (LR)		0.7103	0.6819		0.3717	0.3521		0.8446	0.8065

Table 3: Estimation results for different subsamples of the original dataset, with the innovation distribution being extreme value type 1 $EV1(\mu, \beta)$ (top), and normal $N(0, \sigma)$ (bottom); the "standard" columns refer to innovation densities without normalization, i.e. $EV1(-\gamma, 1)$ and $N(0, 1)$, whereas the "normalized" columns refer to normalized innovation densities, i.e. $EV1(-\gamma\sqrt{1-\rho^2}, \sqrt{1-\rho^2})$ and $N(0, 1-\rho^2)$. L is the value of the log-likelihood function at the solution; p (LR) is the p-value of the likelihood ratio test with $H_0 : \rho = 0$; $\beta = .9999$.

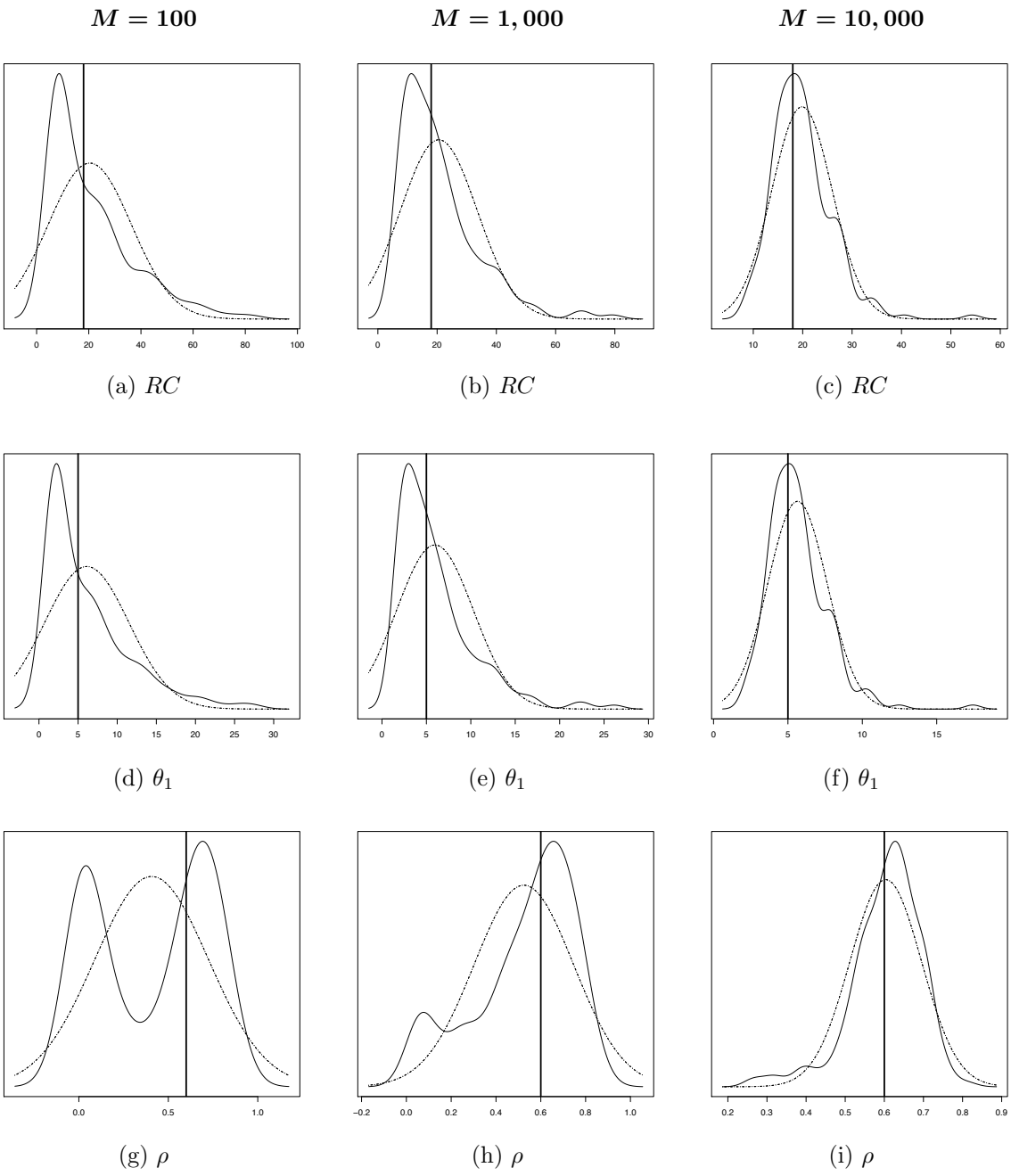


(a) RC (b) RC (c) RC

(d) θ_1 (e) θ_1 (f) θ_1

(g) ρ (h) ρ (i) ρ

Figure 8: Distributions of the maximum likelihood estimates from 200 artificial data sets of different sizes, with the innovation distribution being extreme value type 1 $EV1(-\gamma, 1)$. The bold solid vertical lines denote the true parameter value; the thin solid lines are kernel smoothing estimates of the distributions of the parameter estimates; the dash-dotted lines depict normal distributions with mean and standard deviation of the respective estimates. The lefthand column uses data sets of 100 buses each, the center column 1,000 buses, and the righthand column 10,000 buses.



(a) RC (b) RC (c) RC

(d) θ_1 (e) θ_1 (f) θ_1

(g) ρ (h) ρ (i) ρ

Figure 9: Distributions of the maximum likelihood estimates from 200 artificial data sets of different sizes, with the innovation distribution being standard normal $N(0, 1)$. The bold solid vertical lines denote the true parameter value; the thin solid lines are kernel smoothing estimates of the distributions of the parameter estimates; the dash-dotted lines depict normal distributions with mean and standard deviation of the respective estimates. The lefthand column uses data sets of 100 buses each, the center column 1,000 buses, and the righthand column 10,000 buses.

	$\tilde{\varepsilon} \sim EV1(-\gamma, 1)$						
	True	$M = 100$		$M = 1,000$		$M = 10,000$	
RC	26.0000	11.1135	34.8033	11.0589	32.8574	11.0402	27.7607
		(0.974)	(21.695)	(0.266)	(17.391)	(0.092)	(5.457)
θ_1	7.0000	2.9381	10.2193	2.9023	9.3360	2.9035	7.5885
		(0.435)	(7.381)	(0.121)	(5.789)	(0.041)	(1.748)
RC/θ_1	3.7143	3.8182	3.6105	3.8136	3.6508	3.8027	3.6809
		(0.275)	(0.372)	(0.089)	(0.236)	(0.028)	(0.103)
ρ	0.7000	–	0.6345	–	0.7077	–	0.7069
			(0.241)		(0.081)		(0.035)
p (LR)			0.2380		0.0003		$< 10^{-16}$
			(0.291)		(0.002)		$(< 10^{-16})$
	$\tilde{\varepsilon} \sim N(0, 1)$						
RC	18.0000	7.3097	20.3922	7.1433	20.5688	7.1349	19.8626
		(0.608)	(15.928)	(0.177)	(12.954)	(0.056)	(6.118)
θ_1	5.0000	1.8410	6.1094	1.7589	5.9685	1.7584	5.6418
		(0.268)	(5.391)	(0.077)	(4.295)	(0.025)	(1.990)
RC/θ_1	3.6000	4.0086	3.6162	4.0650	3.6236	4.0581	3.5676
		(0.291)	(0.403)	(0.095)	(0.282)	(0.031)	(0.143)
ρ	0.6000	–	0.4069	–	0.5243	–	0.6035
			(0.317)		(0.222)		(0.091)
p (LR)			0.6946		0.3302		0.0082
			(0.303)		(0.322)		(0.036)

Table 4: Mean and standard deviations of the maximum likelihood estimates from 200 artificial data sets of different sizes, with the innovation distribution being extreme value type 1 $EV1(-\gamma, 1)$ (top), and standard normal $N(0, 1)$ (bottom). The lefthand column uses data sets of 100 buses each, the center column 1,000 buses, and the righthand column 10,000 buses. L is the value of the log-likelihood function at the solution; p (LR) is the mean and the standard deviation of the p-values of the likelihood ratio test with $H_0 : \rho = 0$ carried out on the individual data sets; $\beta = .9999$.

seems to slightly overestimate both cost parameters, the true parameters are always well within one standard deviation. Also, in case of $EV1$ distributed $\tilde{\varepsilon}$ where the overestimation is most apparent, the mean of the estimates clearly gets closer to the true values as I increase the data set size. For the serial correlation parameter ρ, I observe almost perfect recovering of the true parameter value for large data sets, in the $EV1$ case even for moderate data set sizes. Comparing the estimates with serial correlation to the case where serial correlation is ruled out by setting $\rho = 0$, I see that the parameter estimates vary considerably. However, looking at the (probably more relevant) ratio of the cost parameters, I find that the misspecification bias is small in the $EV1$ case, but moderate in the $N(0, 1)$ case. Testing for the statistical significance of the increase in quality of fit by allowing for serially correlated errors using the likelihood ratio test, I find that given a data set of 100 buses (which is comparable to the largest subset of the original data presented above), it is often impossible to reject the no-serial-correlation hypothesis at a

reasonable significance level, even if the true model features serial correlation as in (7). While in the $EV1$ case, significance increases vastly for the larger data sets under consideration, the model with normal $\tilde{\varepsilon}$ has a surprisingly low increase in quality of fit, along with relatively large p-values even for large data sets.

Turning my attention to Figures 8 and 9, I notice that for the smaller data sets the distribution of the parameter estimates is clearly not normal. Moreover, it even appears to be bimodal, with one solution being the no serial correlation case. However, since for the large data sets, the distributions apparently become closer to the density of the normal distribution, especially for the cost parameters in the $N(0,1)$ case, and for the serial correlation parameter in the $EV1$ case.

At this point it is worthwhile commenting on the sources and potential impact of numerical truncation error: First, I found that the likelihood function is quite flat; keeping in mind that the stopping criterion of an optimization algorithm introduces a truncation error of its own, this could well explain the local modes on these tails. Second, while I use a specific Gaussian rule for the normally distributed $\tilde{\varepsilon}$, namely the Gauss–Hermite rule, I use a change of variable to adapt the Gauss–Legendre rule with uniform weighting to the extreme value distribution, which most likely does not perfectly cover the fat tail of this distribution; also, both schemes are applied to a function that is not globally continuously differentiable because of the max-operator. Consequently, in flat regions of the likelihood function, the approximation error of the EV function from both the integration and the interpolation error might dominate the (true) change in the objective function value, making it hard for the solver to distinguish between real progress and computational noise. Thus, it is difficult to judge which effects come from the model structure, and which are numerical artifacts.

5 Conclusion

This paper develops a method to efficiently approximate the (marginal) likelihood function of continuous state hidden Markov models. More precisely, I decompose the integral over the unobserved state variables in the likelihood function into a series of lower dimensional integrals, and successively approximate them using lower dimensional quadrature rules, and interpolation between the time steps; I call this procedure recursive likelihood function integration (RLI), and I provide rigorous error and convergence analysis of the new method as well as assumptions on the model for the theoretical results to be applicable.

As an application, I apply this method to the bus engine replacement model of Rust (1987) featuring serially correlated errors and using the original dataset, finding barely any serial correlation. Also, the parameter estimates vary substantially, compared to the case of serially uncorrelated errors. Second, I verify the RLI algorithm's ability to recover the parameters of the same model in an extensive Monte Carlo study with simulated data sets, finding that the method is indeed able to recover the parameters used for the simulation, particularly in the case of the serial correlation parameter, which is recovered to very high precision.

As I mention in the introductory section, the recursive likelihood function integration is not the only approach to the estimation of DDCMs with serially correlated unobserved state variables. While I cite some recent alternative methods, I do not compare to them in terms

of runtimes, accuracy, or other important metrics. Rather, the goal of this paper was to show that the integration of the serially correlated variables in the computation of the likelihood function can be done with complexity that is linear in the time horizon, making the application of high performance quadrature rules such as Gaussian quadrature well feasible. For a quantitive comparison of the various methods to be insightful, a rigorous experimental design is needed, in order to compare the different aspects of computational efficiency, numerical accuracy, and scaling properties, based on unified models and environments. This in-depth comparison study is subject to future research.

Second, while the theoretical results of this paper directly apply to multi-dimensional (unobserved) state variables, no particular methods for multi-dimensional quadrature and interpolation are applied and tested for their practical performance in the RLI context in this paper. Moreover, the results are limited to Markov processes of order 1. Both topics—while not adding much theoretical insight—are important subjects for future research because of their practical relevance.

A Numerical Methods

A.1 Approximation of the Expected Value Function

This section describes the steps necessary to numerically approximate the expected value as a function of all possible states, as in equation (8).

Numerical integration. In contrast to the case of extreme value type I iid distributed unobserved state variables, no closed form solution to the integral (8) exists; thus, I have to approximate it by numerical quadrature. A variety of methods for multi-dimensional integration exists; see, for example, chapter 7 of Judd (1998) for an overview, or chapter 4 of Press et al. (2007) for an implementation oriented approach. Throughout the paper, I use Gaussian quadrature, which is known to be very efficient for the integration of functions that can be well approximated by a polynomial. While this condition is obviously violated for the value function (because of the kink potentially induced by the max-operator), one can still show Gaussian schemes to be convergent for any Riemann integrable function, and, moreover, they are reported to often outperform other widely used integration schemes, even in the presence of singularities; see Judd (1998) and the literature cited therein. Also, Stinebrickner (2000) successfully applied the Gaussian quadrature rules to expected value function approximation for DDCMs with serial correlation.

The n-node Gaussian quadrature rule approximates

$$\int_a^b f(y)w(y)dy \approx \sum_{i=1}^{n} \omega_i f(y_i) \tag{161}$$

where $w(y)$ is a non-negative weighting function with finite integral (including unity for $|a|, |b| < \infty$). The integration nodes y_i are the roots of the degree n polynomial of the family of polynomials that are mutually orthogonal with respect to weighting function $w(y)$. The corresponding weights ω_i are chosen such that every polynomial of degree $2n-1$ is integrated *exactly*; for the corresponding formulas, see, for example, Kythe and Schäferkotter (2005). Since both nodes and weights should be computed to high accuracy, they are often tabulated for some frequently used families of orthogonal polynomials.

When taking expectations of functions of continuous random variables, the integration problem (161) arises naturally, with the density function being used as weighting function $w(x)$. Obviously, this approach requires the availability of polynomials that are orthogonal with respect to the density function in use. For some distributions, these families are well known, such as the Hermite polynomials for normally distributed random variables. For most other distributions however, the necessary polynomials (and their roots) are unknown, and have to be computed first. Alternatively, one can map the support of the corresponding density function to $[-1, 1]$ by a change of variable,[25] and approximate the resulting integral using the Gaussian rule based on Legendre polynomials, which are orthogonal with respect to the unity weighting function on $[-1, 1]$. Using this procedure, I found that expectations of extreme value distributed random variables can be approximated quite efficiently.

[25] For example, if the inverse of the cumulative distribution of a distribution with density $w(y)$, $W^{-1}(y)$ exists, one can apply the following change of variables: $\int_{-\infty}^{+\infty} f(y)w(y)dy = \int_0^1 f(W^{-1}(y'))dy'$.

Directly approximating (8) by Gaussian quadrature has a potential caveat, since it would require one to find polynomials that are orthogonal with respect to the conditional probability density function, $p_{\varepsilon'}(\varepsilon'|\varepsilon)$, and thus different nodes and weights for each ε. Consequently, I reformulate the integral in (8) in terms of the unconditional probability density function $p_{\tilde{\varepsilon}'}(\tilde{\varepsilon}'(i))$,

$$\iint \max\big\{u(0,x') + \rho\varepsilon(0) + \tilde{\varepsilon}'(0) + \beta EV_\theta(x', (\rho\varepsilon(0) + \tilde{\varepsilon}'(0), \varepsilon'(1)), \tag{162}$$

$$u(1,1) + \tilde{\varepsilon}'(1) + \beta EV_\theta(1, (0, \varepsilon'(1)))\big\}\, p_{\tilde{\varepsilon}'(1)}(d\tilde{\varepsilon}'(1)) p_{\tilde{\varepsilon}'(0)}(d\tilde{\varepsilon}'(0)) \tag{163}$$

and compute (or look up) one single set of nodes and weights for weighting function $p_{\tilde{\varepsilon}'(i)}(\tilde{\varepsilon}'(i))$.

Since the integration in (162) is of dimension $N = 2$,[26] but Gaussian rules are per se one-dimensional, I use them extended to N dimensions by the product rule, which generalizes (161) to N dimensions by

$$\int_{[a,b]^N} f(y^1, \ldots, y^N) \prod_{i=1}^{N} w_i(y^i) d(y^1, \ldots, y^N) \approx \sum_{i_1=1}^{n} \cdots \sum_{i_N=1}^{n} f(y_{i_1}^1, \ldots, y_{i_N}^N) \prod_{j=1}^{N} \omega_{i_j}^j \tag{164}$$

where $f : \mathbb{R}^N \to \mathbb{R}$, $w_i : \mathbb{R} \to \mathbb{R}$ is the weighting function for dimension i, and y_j^i and ω_j^i are the nodes and weights of the corresponding one-dimensional Gaussian rule (indexed by j), applied to dimension i.[27]

Function approximation. Generally, the expected value function is a continuous function of ε, and I need to approximate it as such, but by a finite number of parameters only. Assume for the moment that I can evaluate an unknown function $f(y)$ at arbitrary points. Then, I can choose a set of nodes $y_i \in [a, b]$, and construct an interpolating function $\hat{f}(y)$, such that $f(y_i) = \hat{f}(y_i)\ \forall y_i$. Obviously, I want to choose $\hat{f}(y)$ such that $|f(y) - \hat{f}(y)|$ is "small everywhere", not just at the interpolation nodes y_i. More formally, I want to control the interpolation error $\sup_{y\in[a,b]} |f(y) - \hat{f}(y)|$.

A general, but computationally rather expensive approach to node choice is adaptive procedures: given some interpolant $\hat{f}^{(h)}(y)$, I evaluate the quality of approximation, $|f(y) - \hat{f}^{(h)}(y)|$, at different values of the argument (different from y_i), and I insert new nodes where the approximation quality is poor; then, I construct a new interpolant $\hat{f}^{(h+1)}(y)$ on the set union of old and new nodes. This procedure is iterated until some convergence criterion is met. Adaptive methods are particularly well suited for functions with "difficult" shape properties, for example functions with greatly varying curvature, kinks, or discontinuities, and to explicitly control the approximation error. For the actual interpolation over such a grid, piecewise polynomial interpolation, such as piecewise linear interpolation (PLI) or higher order splines, proved to be a reliable choice.

[26] The dimension of the integration over the unobserved state variable in DDCMs is usually $(N-1)$-dimensional, because the decisions of the agents in the model are driven by utility differences rather than levels. In this case however, since I assume that serial correlation is only present in one dimension of the error, the reformulation of the model in terms of the differences of errors does not reduce dimensionality. Thus, the integration must be carried out over all the N dimensions.

[27] Note that in order to use the product rule (164) to compute expectations, the dimensions of the random variable must be mutually independent. For more general multivariate distributions, see, for example, Jäckel (2005).

Since I want to have direct control over the error of the approximation of EV_θ, I choose an adaptive approximation method; in particular, I want to assure uniform approximation quality for different values of θ, in order to compute the corresponding likelihood function values to high accuracy. Therefore, I employ the method of Grüne and Semmler (2004), which repeatedly refines an interpolation grid until a global approximation error criterion is met. At this point, it is important to note that I cannot directly evaluate the true (but unknown) expected value function EV_θ, because it is only implicitly defined by (9). Fortunately, to discuss this grid adaption method, it is sufficient to assume that the method is supplied with an approximation $\widehat{EV}_\theta^{(h)}(\,\cdot\,;a)$ from the previous iteration of the adaption process, which is now explicitly parametrized by the finite-dimensional vector $a \in \mathbb{R}^A$. Let $\Gamma_\theta^{(h)}$ be the grid at the beginning of iteration h. For each cell[28] c_l of grid $\Gamma_\theta^{(h)}$, I approximate the solution to the following optimization problem:[29]

$$\eta_l = \max_{\varepsilon \in c_l} |\widehat{EV}_\theta^{(h)}(x,\varepsilon;a) - T(\widehat{EV}_\theta^{(h)})(x,\varepsilon;a)| \tag{165}$$

Then, Grüne (1997) showed that the maximum error over all cells, $\eta = \max_l\{\eta_l\}$, defines an approximation error bound by

$$\max_{x\in X,\varepsilon\in R^N} |EV_\theta(x,\varepsilon) - \widehat{EV}_\theta^{(h)}(x,\varepsilon;a)| \leq \eta \frac{1}{1-\beta} \tag{166}$$

where EV_θ represents the true (but unknown) expected value function. The method of Grüne and Semmler (2004) inserts new nodes into those cells c_l where the corresponding error η_l is larger than some threshold. Finally, I construct new interpolant $\widehat{EV}_\theta^{(h+1)}(\,\cdot\,;a)$ on the refined grid $\Gamma_\theta^{(h+1)}$. (In order to parametrize it, I need to solve for the fixed point (9), which I will discuss shortly.) This procedure is repeated until the maximum (global) approximation error $\eta(1-\beta)^{-1}$ is smaller than the desired approximation error, $\bar{\eta}$.

One particular advantage of the method of Grüne and Semmler (2004) is that it not only allows for refinement, but easily extends to grid coarsening, by identifying and removing nodes that do not increase approximation accuracy. Combining coarsening and refinement, I can construct a grid *updating* procedure, which can be integrated with a nested fixed point algorithm (NFXP). In NFXP, the likelihood maximization ("outer loop") repeatedly feeds different values of θ into the expected value function approximation ("inner loop"); thus, rather than building up from scratch an interpolant for each new value of $\theta^{(k+1)}$, it can be obtained from updating an interpolant that has previously been built for some other value $\theta^{(k)}$ (see Section A.2 below).

Note that due to the fact that serial correlation is only allowed in $\varepsilon(0)$, $EV_\theta(x,\varepsilon)$ is constant in $\varepsilon(1)$. Consequently, I only need to approximate it as a one-dimensional function of $\varepsilon(0)$. Therefore, I can use piecewise linear interpolation to construct $\widehat{EV}_\theta$. However, the methodology generalizes to higher dimensions by replacing PLI with multi-dimensional interpolation.

Finally note that, since—in this formulation of the model—mileage has been discretized, I need to approximate EV_θ as a separate continuous function of ε for each mileage state $x \in X$

[28] In this context, cell c_i of an n-dimensional grid Γ is defined as the hypercube spanned by $\{y_j \in \Gamma : y_i^k \leq y_j^k \leq \min_l\{y_l^k : y_i^k < y_l^k\}, k = 1,\ldots,n\}$, where y^k is the kth element (dimension) of the vector y.

[29] Note that since the model is already discretized in terms of mileage state x, finding the maximum error within each cell does not explicitly involve x; rather, one has to carry out the error estimation for all possible mileage states independently.

simultaneously; thus, $\widehat{EV}_\theta(\,\cdot\,;a)$ is really a set of interpolants. If, in contrast, mileage would enter the model as a continuous variable, $\widehat{EV}_\theta(\,\cdot\,;a)$ would rather be a single 2-dimensional interpolant. However, discrete mileage is necessary to nest the original model without serial correlation as a special case.

Non-linear system. The last few paragraphs discussed the choice of a function approximation scheme and interpolation grid creation, but left out how to actually evaluate the unknown function EV_θ, which is only implicitly defined as the fixed point of T. While this fixed point is generally a continuous function, its substitution by an approximating interpolant $\widehat{EV}_\theta(\,\cdot\,;a)$ simplifies the problem to a non-linear system of D equations in A unknowns,

$$\widehat{EV}_\theta(x,\varepsilon;a) = T(\widehat{EV}_\theta)(x,\varepsilon;a) \quad \forall (x,\varepsilon) \in \Gamma_\theta, a \in \mathbb{R}^A \tag{167}$$

where D is the number of elements in Γ_θ, and thus each $(x,\varepsilon) \in \Gamma_\theta$ defines one equation of (167), and the parameters a of the interpolant are the variables. From the parameter vector a^* that solves (167), I can directly construct the interpolant $\widehat{EV}_\theta(\,\cdot\,;a^*)$. This procedure is known as collocation, which is a particular variant of a projection method for the approximation of functions that are defined by functional equations; see Judd (1998), chapter 11. Finally, I compute the approximation error of $\widehat{EV}_\theta(\,\cdot\,;a^*)$ as defined by (166); if it is sufficiently small (smaller than $\bar{\eta}$), I accept my approximation of EV_θ; otherwise, I refine the interpolation grid Γ_θ, and solve (167) for the new grid.

Similar to Rust (1987), I use methods that directly solve the non-linear system

$$\widehat{EV}_\theta(x,\varepsilon;a) - T(\widehat{EV}_\theta)(x,\varepsilon;a) = 0 \quad \forall (x,\varepsilon) \in \Gamma_\theta, a \in \mathbb{R}^A \tag{168}$$

to high accuracy. Given the accuracy needs of my application, Newton (or quasi-Newton) methods are particularly interesting, because they show quadratic (superlinear) convergence close to the solution under some conditions.[30] However, these methods require the evaluation of the Jacobian matrix J of the non-linear system (168), which is generally of size D^2, and thus can be prohibitively expensive to compute for large systems. In particular, given an adaptively refined grid, the size of J can become an issue since the number of equations of (168) is defined by the number of nodes in Γ_θ, and thus the system grows larger as the grid is refined. However, analogously to the original model, if the Markov transition matrix of the discrete states is sparse, J is also sparse; thus, using (quasi-)Newton methods can still be feasible because the number of non-zero elements in the Jacobian grows much more slowly than the number of grid nodes. Figure 10 illustrates the sparseness pattern of my problem.

To numerically solve the fixed point problem (9), I either use the "ipopt" package (Wächter and Biegler, 2005), in conjuction with the "pardiso" sparse linear solver (Schenk and Gärtner, 2004), or the quasi-Newton trust-region method of the R-package "nleqslv" (Hasselman, 2014), depending on the size of the problem.

[30] Loosely speaking, quadratic convergence means that, close to the solution, the number of correct digits of the result roughly doubles in every Newton step. More formally, suppose that for $f : \mathbb{R}^n \to \mathbb{R}^n$, a solution y^* to the system $f(y^*) = 0$ exists, the Jacobian function $J : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is Lipschitz continuous, and the Jacobian matrix at the solution, $J_f(y^*)$ is non-singular. Then, if $y^{(0)}$ is sufficiently close to the solution y^*, the residual decays quadratically for each Newton iteration, thus $\exists K > 0 : \|y^{(k+1)} - y^*\| \leq K\|y^{(k)} - y^*\|^2$.

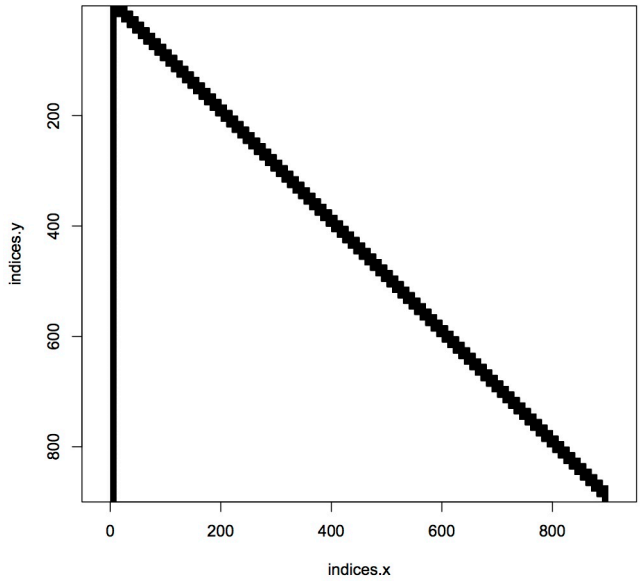


Figure 10: Sparseness pattern of the Jacobian of the non-linear system (168).

Figure 11 plots an example of the expected value function, where each of the black lines represents the expected value as a function of $\varepsilon(0)$, for a particular value x. I want to emphasize again that the procedure to compute an approximation of $EV_\theta(x, \varepsilon)$ as presented in this section easily generalizes to other models, with an arbitrary number of decisions N, and serial correlation in all dimensions of the unobserved state variables, by choosing a multi-dimensional interpolation scheme.

A.2 Likelihood Function Maximization

To approximate the marginal likelihood function of the bus engine replacement model using recursive likelihood function integration, I use Gaussian quadrature as outlined in the previous section (in the context of expected value function approximation). Note that while I write all integrals in this section as integrals over ε for simplicity, I have to reformulate them in terms of $\tilde{\varepsilon}$ by a linear change of variables in order to approximate them by Gaussian quadrature (see Section 2.2.1). Also, for numerical reasons, I chose a slightly different change of variables to map the integration domain from $[-\infty, \infty]$ to $[-1, 1]$, (see Judd, 1998, p. 204). Furthermore, I use Akima splines (Akima, 1970) to approximate the integral over ε_t as a function of ε_{t-1}.

Obtaining the maximum likelihood estimate of θ, given data $\{x_t, i_t\}_{t=0}^T$, requires us to find a solution to the following two problems *simultaneously*:

$$\hat{\theta} = \arg\max_\theta \; L(\,\theta \,|\, \{x_t, i_t\}_{t=0}^T, \widehat{EV}_\theta) \tag{169}$$

$$\widehat{EV}_\theta(x, \varepsilon; a) = T(\widehat{EV}_\theta)(x, \varepsilon; a) \quad \forall (x, \varepsilon) \in \Gamma_\theta, a \in \mathbb{R}^A \tag{170}$$

While there exist methods that directly solve (169) and (170) simultaneously as a constrained optimization problem, namely the mathematical programming with equilibrium constraints (MPEC) approach to DDCM estimation of Su and Judd (2012), I use the well known nested

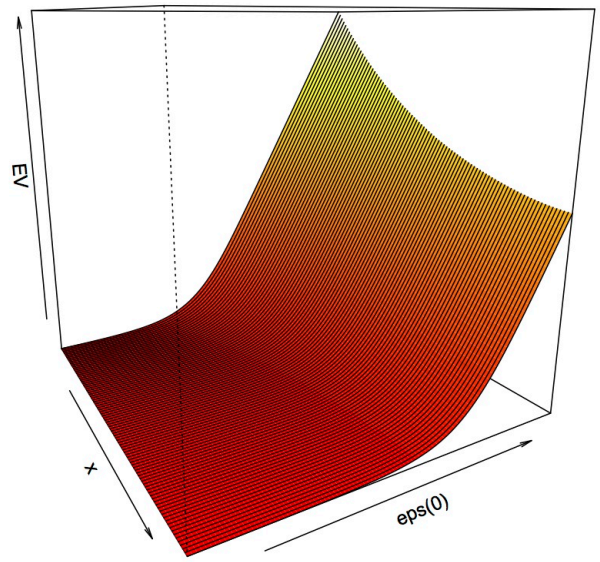


Figure 11: The expected value function $EV_\theta(x, \varepsilon)$ for $\rho = 0.6$, $RC = 14$, $\theta_1 = 2$, and the innovation distribution being $EV1(-\gamma, 1)$.

fixed point (NFXP) approach of Rust (1988).[31] In NFXP, the likelihood maximization is performed as a repeated two step procedure: First, given a parameter guess $\theta^{(k)}$, one computes the expected value function $EV_{\theta^{(k)}}$ as a fixed point of operator T by solving (170). Second, one evaluates the likelihood function for $\theta^{(k)}$, using the approximation of $EV_{\theta^{(k)}}$ just previously obtained. The optimization algorithm then constructs a new parameter guess $\theta^{(k+1)}$, and the procedure starts again by approximating $EV_{\theta^{(k+1)}}$. This is iterated until convergence of the maximization algorithm.[32] Thus, (169) can be solved as an unconstrained problem.

Recall that the interpolation grid $\Gamma_{\theta^{(k)}}$, over which the corresponding approximating interpolant $\widehat{EV}_{\theta^{(k)}}(\cdot\,; a)$ satisfies some error bound $\bar{\eta}$, depends on $\theta^{(k)}$. Thus, each step of the maximization routine, from $\theta^{(k)}$ to $\theta^{(k+1)}$, requires one to iteratively update the grid from $\Gamma_{\theta^{(k)}}$ to $\Gamma_{\theta^{(k+1)}}$, until the maximum approximation error of $\widehat{EV}_{\theta^{(k)}}(\cdot\,; a)$ is bounded by $\bar{\eta}$ again; this procedure ensures that for each likelihood function evaluation, the approximation error of the corresponding expected value function is controlled.[33]

[31] The MPEC approach to DDCM estimation of Su and Judd (2012) "combines" the solution of the fixed point and the maximization of the likelihood by solving the original constraint formulation of the likelihood maximization problem (169). This procedure is considered to be more efficient in some cases, because it does not require one to solve the fixed point equation (9) for each parameter guess, even if it is far away from the solution; rather, it imposes the fixed point condition to hold only at the solution. However, directly integrating MPEC with adaptive interpolation grids creates two potential problems: First, adding a grid node corresponds to adding a constraint to the optimization problem, while the optimization algorithm runs. Second, adaptive methods usually require the approximation of an iteration to be completed in order to compute the approximation quality for the insertion decision, which in my case is not possible until (9) has been solved, which in turn contradicts the MPEC idea.

[32] Since the fixed point of T is usually obtained using an iterative method, solving the dynamic problem is often referred to as the "inner loop" in this context, while the maximization procedure is referred to as the "outer loop".

[33] Controlling the maximum approximation error does not imply that it is constant over the maximization procedure. Rather, I choose $\bar{\eta}^{(k)}$ to be decreasing in the iterations of the optimizer, in order to compute the fixed point to lower accuracy far away from the solution, but to high accuracy close to it.

Algorithm 2 summarizes the nested fixed point algorithm to solve (169).

Algorithm 2 Nested fixed point algorithm with adaptive grid updating.

```
initialize θ, Γ_θ, a
while not converged do
    while η(1 − β)^{−1} > η̄ do
        solve ÊV_θ(x, ε; a) = T(ÊV_θ)(x, ε; a)   ∀(x, ε) ∈ Γ_θ, a ∈ ℝ^A
        update Γ_θ (coarsening and refinement)
    end while
    evaluate L(θ)
    compute next θ
end while
```

For the model under consideration, the maximization of the likelihood function is a non-linear, partially box-constrained optimization problem with three free parameters. To numerically solve this problem, I employ the model-based, derivative-free trust-region method "bobyqa" (Powell, 2009).[34]

B Open Source Software

This appendix lists all open source software packages used to obtain the results presented in this paper, including version information.

The main framework used to implement the method of this paper is R, version 3.0.3 (R Core Team, 2014). Time critical components are implemented in C++, and interfaced to R using the "Rcpp" package, v0.11.1 (Eddelbuettel and François, 2011). The code is parallelized on the C++ level using openMP. All interpolation on the C++ level is carried out using the respective routines of the GNU Scientific Library, v1.16 (Galassi et al., 2014). Gaussian quadrature nodes are computed using the R-packages "fastGHQuad", v0.1-1 (Blocker, 2011), and "pracma", v1.6.4 (Borchers, 2014). Distribution functions, quantile functions, and random number generators for the extreme value distribution are provided by the R-package "evd", v2.3-0 (Stephenson, 2002). To numerically solve the fixed point problem (9), I use the "ipopt" package, v3.11.7 (Wächter and Biegler, 2005), in conjuction with the "pardiso" sparse linear solver, v5.0.0 (Schenk and Gärtner, 2004), interfaced by the R-package "ipoptr", v0.8.4, by Jelmer Ypma (which is distributed as part of the ipopt package), and the quasi-Newton trust-region method of the R-package "nleqslv", v2.1.1 (Hasselman, 2014). For the likelihood maximization problem, I employ "bobyqa" (Powell, 2009), interfaced by the "minqa" R-package, v1.2.3 (Bates et al., 2012).

[34] According to Powell (2009), the name "bobyqa" is an acronym for "Bound Optimization BY Quadratic Approximation."

References

Aguirregabiria, V. and Mira, P. (2010). Dynamic Discrete Choice Structural Models: A Survey. *Journal of Econometrics*, 156(1):38–67.

Akima, H. (1970). A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures. *Journal of the ACM*, 17(4):589–602.

Arcidiacono, P. and Ellickson, P. B. (2011). Practical Methods for Estimation of Dynamic Discrete Choice Models. *Annual Review of Economics*, 3(1):363–394.

Arcidiacono, P. and Miller, R. A. (2011). Conditional Choice Probability Estimation of Dynamic Discrete Choice Models with Unobserved Heterogeneity. *Econometrica: Journal of the Econometric Society*, 79(6):1823–1867.

Bates, D., Mullen, K. M., Nash, J. C., and Varadhan, R. (2012). *minqa: Derivative-Free Optimization Algorithms by Quadratic Approximation*. R package version 1.2.3.

Blevins, J. R. (2016). Sequential Monte Carlo Methods for Estimating Dynamic Microeconomic Models. *Journal of Applied Econometrics*, 31(5):773–804.

Blocker, A. W. (2011). *fastGHQuad: Fast Rcpp Implementation of Gauss–Hermite Quadrature*. R package version 0.1-1.

Borchers, H. W. (2014). *pracma: Practical Numerical Math Functions*. R package version 1.6.4.

Cai, Y. and Judd, K. L. (2013). Advances in Numerical Dynamic Programming and New Applications. In Schmedders, K. and Judd, K. L., editors, *Handbook of Computational Economics*, pages 479–516. Newnes.

Connault, B. (2016). Hidden Rust Models.

Cosslett, S. R. and Lee, L.-F. (1985). Serial Correlation in Latent Discrete Variable Models. *Journal of Econometrics*, 27(1):79–97.

Davis, P. J. and Rabinowitz, P. (1984). *Methods of Numerical Integration*. Academic Press.

Eddelbuettel, D. and François, R. (2011). Rcpp: Seamless R and C++ Integration. *Journal of Statistical Software*, 40(8):1–18.

Elliott, R. J., Aggoun, L., and Moore, J. B. (2008). *Hidden Markov Models: Estimation and Control*, volume 29. Springer.

Galassi, M., Davies, J., Theiler, J., Gough, B., Jungman, G., Alken, P., Booth, M., and Rossi, F. (2014). *GNU Scientific Library Reference Manual*. version 1.16.

Grüne, L. (1997). An Adaptive Grid Scheme for the Discrete Hamilton-Jacobi-Bellman Equation. *Numerische Mathematik*, 75(3):319–337.

Grüne, L. and Semmler, W. (2004). Using Dynamic Programming with Adaptive Grid Scheme for Optimal Control Problems in Economics. *Journal of Economic Dynamics and Control*, 28(12):2427–2456.

Hasselman, B. (2014). *nleqslv: Solve Systems of Non-Linear Equations*. R package version 2.1.1.

Hotz, V. J. and Miller, R. A. (1993). Conditional Choice Probabilities and the Estimation of Dynamic Models. *The Review of Economic Studies*, 60(3):497.

Imai, S., Jain, N., and Ching, A. (2009). Bayesian Estimation of Dynamic Discrete Choice Models. *Econometrica: Journal of the Econometric Society*, 77(6):1865–1899.

Jäckel, P. (2005). A Note on Multivariate Gauss-Hermite Quadrature. Technical report.

Judd, K. L. (1998). *Numerical Methods in Economics*. The MIT Press.

Kay, S. M. (1983). Recursive Maximum Likelihood Estimation of Autoregressive Processes. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 31(1):56–65.

Keane, M. P., Todd, P. E., and Wolpin, K. I. (2011). The Structural Estimation of Behavioral Models: Discrete Choice Dynamic Programming Methods and Applications. In Ashenfelter, O. and Card, D., editors, *Handbook of Labor Economics*, pages 331–461. Elsevier.

Keane, M. P. and Wolpin, K. I. (1994). The Solution and Estimation of Discrete Choice Dynamic Programming Models by Simulation and Interpolation: Monte Carlo Evidence. *The Review of Economics and Statistics*, 76(4):648–672.

Kress, R. (1998). *Numerical Analysis*. Springer.

Kythe, P. K. and Schäferkotter, M. R. (2005). *Handbook of Computational Methods for Integration*, volume 1. CRC Press.

Larsen, B. J., Oswald, F., Reich, G., and Wunderli, D. (2012). A Test of the Extreme Value Type I Assumption in the Bus Engine Replacement Model. *Economics Letters*, 116(2):213–216.

McFadden, D. (1974). Conditional Logit Analysis of Qualitative Choice Behavior. In Zarembka, P., editor, *Frontiers in Econometrics*, pages 105–142. Academic Press.

McFadden, D. (1981). Econometric Models for Probabilistic Choice. In Manski, C. F. and McFadden, D., editors, *Structural Analysis of Discrete Data with Econometric Applications*, pages 198–272. The MIT Press.

Miller, R. A. (1984). Job Matching and Occupational Choice. *The Journal of Political Economy*, 92(6):1086–1120.

Norets, A. (2009). Inference in Dynamic Discrete Choice Models with Serially Correlated Unobserved State Variables. *Econometrica: Journal of the Econometric Society*, 77(5):1665–1682.

Norets, A. (2012). Estimation of Dynamic Discrete Choice Models Using Artificial Neural Network Approximations. *Econometric Reviews*, 31(1):84–106.

Pakes, A. (1986). Patents as Options: Some Estimates of the Value of Holding European Patent Stocks. *Econometrica: Journal of the Econometric Society*, 54(4):755–784.

Powell, M. J. D. (2009). The BOBYQA Algorithm for Bound Constrained Optimization without Derivatives. Technical report, University of Cambridge.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 3 edition.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing.

Rust, J. (1987). Optimal Replacement of GMC Bus Engines: An Empirical Model of Harold Zurcher. *Econometrica: Journal of the Econometric Society*, 55(5):999–1033.

Rust, J. (1988). Maximum Likelihood Estimation of Discrete Control Processes. *SIAM Journal on Control and Optimization*, 26(5):1006–1024.

Rust, J. (1996). Numerical dynamic programming in economics. In Amman, H. M., Kendrick, D. A., and Rust, J., editors, *Handbook of Computational Economics*, pages 619–729. Elsevier.

Schenk, O. and Gärtner, K. (2004). Solving Unsymmetric Sparse Systems of Linear Equations with PARDISO. *Future Generation Computer Systems*, 20(3):475–487.

Stephenson, A. G. (2002). evd: Extreme Value Distributions. *R News*, 2(2).

Stinebrickner, T. R. (2000). Serially Correlated Variables in Dynamic, Discrete Choice Models. *Journal of Applied Econometrics*, 15(6):595–624.

Su, C.-L. and Judd, K. L. (2012). Constrained Optimization Approaches to Estimation of Structural Models. *Econometrica: Journal of the Econometric Society*, 80(5):2213–2230.

Trefethen, L. N. (2013). *Approximation Theory and Approximation Practice*. Siam.

Wächter, A. and Biegler, L. T. (2005). On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming. *Mathematical Programming*, 106(1):25–57.

Wolpin, K. I. (1984). An Estimable Dynamic Stochastic Model of Fertility and Child Mortality. *The Journal of Political Economy*, 92(5):852–874.